NOAA NESDIS
CENTER for SATELLITE APPLICATIONS and RESEARCH

GOES-R Advanced Baseline Imager (ABI) Algorithm Theoretical Basis Document For Hurricane Intensity

Timothy Olander, UW/CIMSS
Christopher Velden, UW/CIMSS
Jaime Daniels, NOAA/NESDIS/STAR

Version 2.5
July 30, 2012

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ABI	- Advanced Baseline Imager
ADRS	- Ancillary Data Relay System
ADT	- Advanced Dvorak Technique
AIT	- Algorithm Implementation Team
AODT	- Advanced Objective Dvorak Technique
ASCII	- American Standard Code for Information Interchange
ATBD	- Algorithm Theoretical Basis Document
ATCF	- Automated Tropical Cyclone Forecasting system
AVHRR	- Advanced Very High Resolution Radiometer
AWG	- Algorithm Working Group
BD	- Basic Dvorak
CDO	- Central Dense Overcast
CDR	- Critical Design Review
CI	- Current Intensity
CIMSS	- Cooperative Institute for Meteorological Satellite Studies
CIRA	- Cooperative Institute for Research in the Atmosphere
CPHC	- Central Pacific Hurricane Center
CRT	- Cloud Region Temperature
CWRN	- Coldest-Warmest Ring Number
EIR	- Enhanced Infrared
ERT	- Eye Region Temperature
F&PS	- Functional and Performance Specifications
FFT	- Fast Fourier Transform
GEOCAT	- Geostationary Cloud Algorithm Testbed
GNU	- GNU's Not UNIX
GOES	- Geostationary Operational Environmental Satellite
GS	- Ground Segment
HIE	- Hurricane Intensity Estimation
JTWC	- Joint Typhoon Warning Center
MODIS	- Moderate Resolution Imaging Spectroradiometer
MRD	- Mission Requirements Document
MSG	- Meteosat Second Generation
MSLP	- Mean Sea Level Pressure
NHC	- National Hurricane Center
NOAA	- National Oceanic and Atmospheric Administration
ODT	- Objective Dvorak Technique
PORD	- Performance and Operational Requirements Document
RF	- Ring Fitting
RMSE	- Root Mean Squared Error
RMW	- Radius of Maximum Wind
RSMC	- Regional Specialized Meteorological Center
SABI	- Simulate ABI
SC	- Spiral Centering
SDT	- Subjective Dvorak Technique

SEVIRI	- Spinning Enhanced Visible and Infrared Imager
STAR	- Center for Satellite Applications and Research
TC	- Tropical Cyclone
TCFC	- Tropical Cyclone Forecast Center
TRR	- Test Readiness Review
USGS	- United States Geological Survey
UTC	- Universal Time Coordinates
UW	- University of Wisconsin
VAGL	- Vendor Allocated Ground Latency
WMO	- World Meteorological Organization

Abstract

This document is the Algorithm Theoretical Basis Document (ATBD) for the Hurricane Intensity Estimation (HIE) algorithm. It provides a description and the physical basis for the estimation of tropical cyclone intensity using infrared (IR) temperature values measured by the Advanced Baseline Imager (ABI) to be flown on the next generation of Geostationary Operational Environmental Satellite (GOES-R). The HIE has its origins with the Advanced Dvorak Technique (ADT) algorithm that is currently utilized operationally at several tropical cyclone forecast centers worldwide. This document contains a description of the algorithm, including scientific aspects and practical considerations.

1 INTRODUCTION

1.1 Purpose of This Document

The Hurricane Intensity Estimation (HIE) Algorithm Theoretical Basis Document (ATBD) provides detailed information regarding the Advanced Dvorak Technique (ADT), an algorithm that will derive estimates of tropical cyclone (TC) intensity using satellite imagery from the Advanced Baseline Imager (ABI) flown on the GOES-R series of NOAA geostationary meteorological satellites. This document summarizes the HIE/ADT algorithm methodology (through version 4.0) and usage of the application.

The central purpose of this ATBD is to facilitate development of operational Product Generation software for the HIE product which is implemented within the GOES-R Ground Segment product generation subsystem.

1.2 Who Should Use This Document

This document should be referenced by users who wish to operate the HIE, understand the internal workings of the algorithm, and learn how to interpret the analysis output.

1.3 Inside Each Section

This document is broken down into the following main sections.

- **Algorithm Overview**: Describes the development and theory basis for the HIE. This is covered in Section 2.
- **Algorithm Usage**: Provides a detailed description of the algorithm operation, the required input files and control parameters, and the program output. These are covered in Sections 3, 4, and 5.

- **Algorithm Limitations**: Discusses the current limitations of the algorithm and potential paths for future research to improve the HIE. This is covered in Section 6.

1.4 Related Documents

The HIE can utilize input and output data files in Automated Tropical Cyclone Forecasting System (ATCF) format. A complete document related to this formatting style can be found online at the following address:
http://www.nrlmry.navy.mil/atcf_web/

1.5 Revision History

Version 0.1 of this document was created by Timothy Olander of the University of Wisconsin-Madison/Cooperative Institute for Meteorological Satellite Studies (UW-CIMSS). Version 0.2 was also written by Tim Olander at UW-CIMSS after review by Jaime Daniels of NOAA/STAR. Version 0.3 and 1.0 were completed after review and acceptance from AIT for the "Draft Version" document delivery. The HIE ATBD Version 1.0, was amended by Tim Olander at UW-CIMSS in order to update the guide for the "80% Version" document delivery after several significant milestones were achieved with the HIE algorithm, including the Algorithm Version 3 release to AIT, the Critical Design Review (CDR), and Test Readiness Review (TRR). Jaime Daniels supplied substantial review of ATBD Version 1.0 prior to Version 1.1 update, with Chris Velden (UW-CIMSS) providing a significant review of the updated Version 1.0 and 1.1 documents prior to release. Version 1.4 was modified/released to update the HIE ATBD in order to conform to AIT ATBD format standards. Version 1.5, updated the document to include new information, update validation studies, and address questions by Harris reviewers. This document, Version 2.0, addresses all comments from the System Engineering 80% review and final AIT ATBD review prior to the release of the HIE ATBD (100% delivery) Version 2.0

• Version 0.1	Initial CIMSS release	March 2008
• Version 0.2	Modifications/Jaime Daniels review	May 2008
• Version 0.3	Modifications/AIT (Larisa Koval) review	June 2008
• Version 1.0	AIT release (80% delivery)	September 2008
• Version 1.1	Update/Chris Velden review	May 2009
• Version 1.2	AIT release/Larisa Koval review	June 2009
• Version 1.3	Modifications/CIMSS review	July 2009
• Version 1.4	Modifications/Jaime Daniels review	September 2009
• Version 1.5	Modifications/Jaime Daniels review	June 2010
• Version 2.0	Modifications/AIT review (100% delivery)	September 2010
• Version 3.0	Update after ADD/AER code reviews	August 2012

2 OBSERVING SYSTEM OVERVIEW

2.1 Products Generated

The HIE algorithm generates real-time tropical cyclone (TC) intensity estimates using infrared window channel satellite imagery. The primary products generated by the HIE include the current intensity estimate analysis, in terms of wind speed, mean sea level pressure, and Current Intensity number (CI#), from the satellite image being interrogated as well as additional time-averaged and other time-dependent estimate values utilizing specific rules to limit the growth/decay of the intensity estimates over time. Graphical and textual listings of the historical HIE TC estimates can be viewed, saved, and distributed. It must be noted that the wind speed estimate is the only value output by the HIE algorithm which must meet the F&PS reproducibility requirements. Table 1 outlines the specifications for the hurricane intensity product GOES-R as defined in the latest version of the GOES-R Ground Segment Project Functional and Performance Specification (F&PS) requirements document.

Hurricane Intensity	**Specification**
Geographic Coverage	Full Disk
Vertical Resolution	N/A
Horizontal Resolution	2 km
Mapping Accuracy	1 km
Measurement Range	Dvorak hurricane intensity scale values of 1.5 – 8 or leading to wind speeds of 12.8 m/s (25 knots) to 87.5 m/s (170 knots)
Measurement Accuracy	6.0 m/s over the ocean
Refresh Rate/Coverage Time (Mode 3)	30 minutes
Refresh Rate (Mode 4)	30 minutes
VAGL (Mode 3 or 4)	806s
Measurement Precision	8.0 m/s over the ocean
Product Qualifiers	
Temporal Coverage	Day and night
Product Extent	Quantitative out to at least 65 degrees LZA and qualitative beyond
Cloud Cover Conditions	Clear conditions down to feature of interest associated with threshold accuracy
Product Statistics	Over hurricane cases

Table 1: Requirements for Hurricane Intensity product

2.2 Instrument Characteristics

Table 2 below shows the ABI channels and the channel used by the HIE algorithm. The performance of the HIE can be sensitive to imagery artifacts or instrument noise. Calibrated observations are critical because the HIE relies upon measured brightness temperature measurements to determine the TC intensity estimates. Properly navigated imagery is also essential, especially if run in a completely automated fashion using interpolated operational TC forecast positions as a first guess for the storm center determination schemes. The ABI channel specifications are given in the Mission Requirements Document (MRD) section 3.4.2.1.4.0 and it is assumed that the ABI performance will meet these specifications.

Channel Number	*Wavelength (µm)*	*Used in HIE*
1	0.47	
2	0.64	
3	0.86	
4	1.38	
5	1.61	
6	2.26	
7	3.9	
8	6.15	
9	7.0	
10	7.4	
11	8.5	
12	9.7	
13	10.35	✓
14	11.2	
15	12.3	
16	13.3	

Table 2: Channel numbers and wavelengths for the ABI and those used by the HIE.

3 ALGORITHM DESCRIPTION

The Hurricane Intensity Estimation (HIE) algorithm is a program based upon the the Advanced Dvorak Technique (ADT) algorithm (Olander and Velden, 2007). The ADT is a computer-based technique, developed at the University of Wisconsin-Madison/Cooperative Institute for Meteorological Satellite Studies (UW/CIMSS), to objectively determine tropical cyclone (TC) intensity using geostationary satellite infrared imagery. The ADT can be used to classify storm intensity from storm formation, through development and maturation, to dissipation.

The ADT is patterned after the Subjective Dvorak Enhanced Infrared 'EIR' Technique (SDT) (Dvorak, 1975, 1984) which made use of various pattern identification schemes and rules to determine TC intensity. The ADT was originally developed to closely mimic the SDT methodology in terms of intensity determination protocol and the incorporation of various rules and analysis methods. Some of the original SDT rules have subsequently been modified in the ADT as determined by in-depth statistical analysis of ADT performance during application. In addition, selected intensity determination modules within the ADT have adopted regression based methods, further advancing the ADT beyond the scope of the original SDT procedures and capabilities, as well as previous versions of the ADT; the Objective Dvorak Technique (ODT) (Velden et al., 1998), and Advanced Objective Dvorak Technique (AODT) (Olander et al., 2004 and 2002) at UW-CIMSS.

The significant modifications and additions made to the ADT results in an algorithm that bears little if any resemblance to its forerunners in terms of methodology, functionality, and content. The primary modifications from previous objective Dvorak-based methods include the addition of a history file (containing previous intensity estimates obtained during a storm lifecycle), utilization of a time-weighted averaging scheme for the final intensity, new definitions and determinations of environmental cloud-top temperature values used to determine scene type and intensity, the implementation and adjustment of SDT rules governing the variability of the intensity estimates, and the modification of the basic relationships governing the intensity estimates away from the pattern recognition-type SDT methodologies towards the use of regression-based equations utilizing objectively analyzed environmental parameters. These changes have led to more stable and statistically sound estimates of intensity by the objective ADT scheme.

3.1 Algorithm Overview

The ADT/HIE algorithm can perform an intensity analysis of any active global TC with an intensity of Tropical Depression status and a maximum wind of >25kts. Identification of storm systems for which to initialize and utilize the ADT algorithm are determined by official WMO Tropical Cyclone Forecast Center (TCFC) bulletins. For each TC, the ADT algorithm is intended to execute in an automated fashion, through the use of UNIX shell scripts via an automated command scheduler or other similar system setup, over the entire life cycle of the TC until the TCFC bulletins are discontinued by the issuing forecast center. The processing logic used to determine an "active TC" is conducted prior to and external from the actual execution of the ADT itself, so this process is not described in this document. It is left to the implementing organization/group of the ADT to implement the actual process of determining an "active TC". The ADT will assume that the image being examined contains an active TC of some valid intensity.

The primary advantage the ADT presents to TC forecasters is the ability to objectively determine TC intensity and storm center location without human subjectivity issues, or any manual intervention unless specifically requested. Initial TC storm center positioning is primarily determined via a polynomial interpolation of a short-term TC position forecast provided by the responsible TCFCs worldwide. Once determined, the infrared satellite image is interrogated by a series of algorithms to search for a potentially more accurate TC storm center based upon analysis of the cloud top temperature field.

Once the TC storm center has been determined, the "scene type" is automatically selected using a series of cloud top temperature statistical analysis routines. The scene type governs the subsequent regression schemes to be employed by the ADT to estimate intensity, and is based on cloud pattern recognition. Having established the scene type, the current TC intensity estimate is then derived for the image being investigated. The storm intensity estimate is output as maximum sustained 1-minute surface winds (Kts), and mean sea level pressure (MSLP) at the center (hPa). Various rules governing the intensity change rate, along with a time-averaging routine (using previous intensity estimates stored in an external storm file), are applied during individual stages of the intensity determination process, and are designed to limit/smooth the evolution of the TC intensity over time. The maximum wind and MSLP values are determined from an empirical relationship between the final calculated "current intensity" (CI#) estimate and the intensity parameters, and is TC basin dependent. A final minor adjustment to the MSLP is based upon the current latitude of the storm (Kossin and Velden 2005).

All of the critical parameters utilized in the determination of the intensity estimate values within the ADT operation are saved to an external file called the ADT "History File". These values, stored for all previous analyses, are used by the ADT algorithm for various internal logic procedures as well as in the determination of specific parameters critical to the intensity estimation process. They can also be accessed by an analyst seeking specific information on past estimates.

3.2 Processing Outline

The top level flowchart describing the HIE processing is presented in Figure 1. Many of the functions displayed in Figure 1 are discussed in greater detail in the various Mathematical Description discussions in Section 3.4.2.

The automated derivation of the HIE intensity estimates follows the path indicated in Figure 1 below, with various branches governing non-intensity derivation branches controlled via input values entered at the start of the algorithm processing. These non-intensity derivation inputs allow various functions such as listing, graphing, or editing the HIE history file (known as "tailored products" for the HIE). The execution of the HIE will be performed in a completely automated fashion from a scheduled command on the platform where the HIE is installed and executed without any human interaction.

During the operation of the HIE algorithm, the program will need to automatically determine the correct position of the storm center (Section 3.4.2.1). Once this position is obtained, the derivation of the cloud pattern "scene type" will be performed (Section 3.4.2.2). The derivation of the scene type will in turn determine the method for determining the intensity estimate (Section 3.4.2.4) for the scene being interrogated. The derived intensity estimate will then be output to the user (Section 3.4.3) and saved to the HIE history file (Section 3.3.2.1).

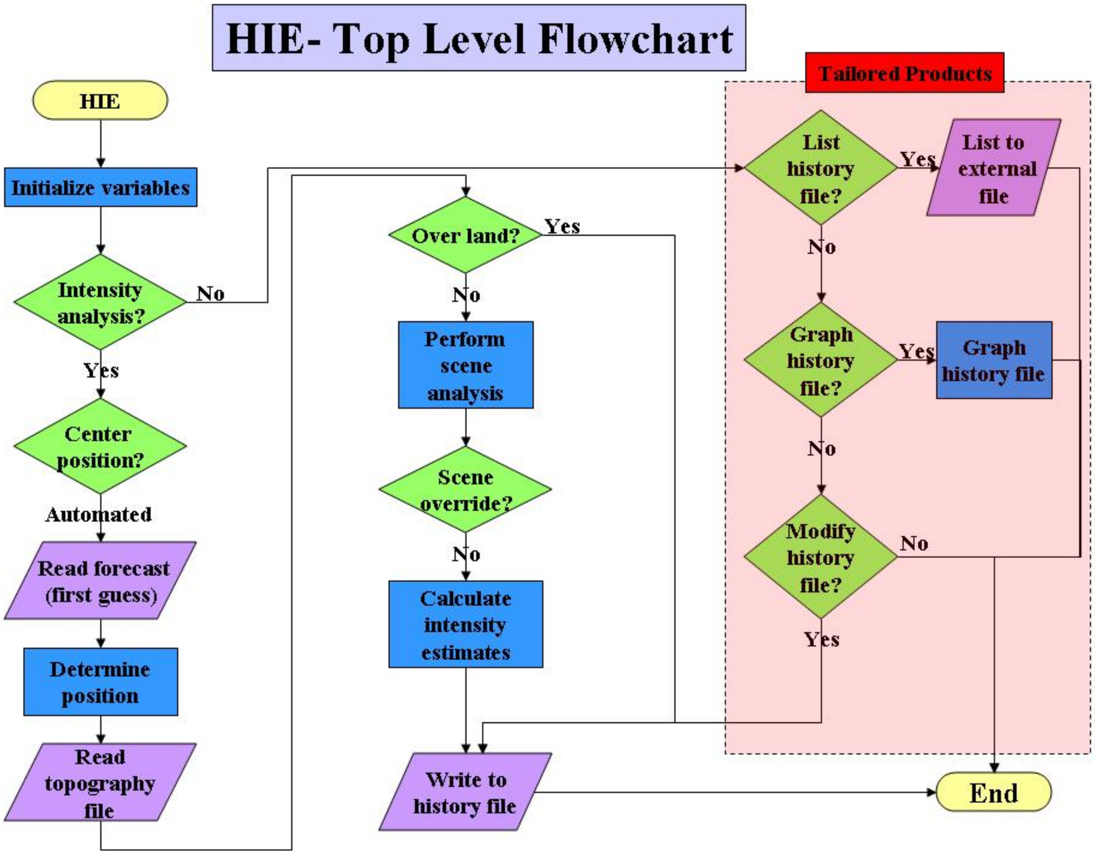

Figure 1: HIE Top-Level processing flowchart.

To utilize the HIE, the first step involves an initialization to set up the file and directory structures needed for I/O, storage, and ancillary data access. The details of this process are beyond the scope of this document, but the needed elements will be posted in a User's Guide. Inputs consist of various keywords and parameter entries from a command line which control the Top Level path to be followed (green diamonds). These inputs can be automated via external scripts set up by the user in a fully automated system. If an intensity estimation is to be calculated through a fully automated procedure, the HIE must perform three main tasks; 1. Determine the TC center position, 2. Perform an analysis to select the proper scene type, and 3. Calculate the intensity estimate for the selected scene type. These processes are described in the following sections. Non-intensity related tasks can also be automated to perform task such as listing of the history file or producing other output files (e.g. ATCF listings).

Input and output data requirements (purple trapezoids) will be discussed in the final subsection following the intensity estimate discussion.

3.3 Algorithm Input

Filename	Format	Originates	Contents
Variable	Binary	Internal	History file (HIE analysis parameters) netCDF format
Variable	Binary	NOAA	Infrared (longwave IR) Satellite Image
User determined	ASCII	Many	Tropical cyclone forecast
Digelev_hires_le.map	Binary	NOAA	.1° resolution topography flags
Variable	ASCII	NESDIS	Passive Microwave "Eye Score" value
Variable	ASCII	Many	Courtney/Knaff/Zehr wind-pressure TC environment values

Table 3: Input data files for HIE algorithm.

3.3.1 Primary Sensor Data

The HIE currently utilizes longwave infrared window (approximately 10.7µm central wavelength channel; band 4 on the current GOES platforms) imagery from any of the operational geostationary satellite platforms, or from the MODIS instrument on the polar-orbiting NOAA Aqua or Terra satellites. The imagery can be in native satellite projection or remapped Mercator projection however they must contain proper navigation and calibration information. For GOES-R, Band 13 (10.35 µm) data will be supplied to the HIE from the AIT framework which obtains the data via the Level 1B Processing System. AIT software library routines will interface directly with the HIE algorithm to transfer the necessary satellite data information to the intensity estimation routine as needed.

It is assumed that the entire storm and TC analysis region (TCFOV, defined below) will be within the field of view of the satellite image being examined, centered at the automated storm center position, and will also contain a limited number of bad scan lines and pixels per line. Typical temperature ranges for the IR pixels should be between approximately -100°C and +30°C. The HIE will check for bad pixels and lines in the TCFOV and replace them as necessary with either the adjoining (western) pixel values or adjoining (previous) line values. A bad pixel is defined as a temperature value greater than or equal to 320°K or less than 120°K. If more than 10 pixels on each line within the TCFOV are replaced the line is flagged as bad. If more than 10 bad lines are found in the TCFOV, the TCFOV is deemed to be too corrupted for use and the HIE algorithm will end. An error message and flag value (Section 5.4) will be output to indicate that this has occurred.

The imagery can contain data over ocean and land surfaces. The TCFOV in the HIE algorithm for the scene type and intensity derivation routines is a circle of radius 136km, so a box of size 136 X 136 contains the TCFOV (assuming a spatial resolution of 2km in the infrared window channel with GOES-R ABI data). The analysis region for the automated storm centering routines within the HIE algorithm package a bit larger than the scene type/intensity analysis domain at 190km plus 5 extra pixels on each side of the IR data array box. The size of the TCFOV array box containing the IR data for the automated storm center determination processing and scene type/intensity determination schemes are dependent upon the line and element resolution of the data in the IR image. For example, for a 2km image, the line minima/maxima are defined:

Min/Max Line = Center_Line +/- (Analysis_Radius/Line_Resolution) + 5 pixels

If the minimum or maximum lines or elements go beyond the extent of the IR image being examined, the analysis region will be adjusted/shifted to contain IR image data (not read off the edge of the image), if possible. For example, if the analysis region extends 20 pixels past the maximum line value of the IR image being examined, the maximum value will be set to last line value available and the minimum value will be reduced by 20 pixels to accommodate the shift in the analysis region. If the analysis IR image is too small to accommodate this shift, an error will be returned and terminate the execution of the HIE.

3.3.2 Ancillary Data

3.3.2.1 HIE History File

The HIE history file will be a binary file in netCDF-format containing HIE intensity estimates, locations, and other specific information for each image analysis. The HIE algorithm utilizes records in the history file in the determination of the intensity estimate values as well as the application of various rules utilized within the algorithm. This file will be ‘initialized’ the first time the HIE is activated on a target TC, either automatically (with the T# intensity value that the HIE determines for the first image), with a user defined input T# value for intensity (most likely the T# corresponding to the estimated wind speed from the TCFC forecast/discussion file, as shown in Section 3.3.2.2), or a default T# value of 1.0, dependent upon however the program is set up to be executed automatically. Each analysis stored in the history file will contain the following netCDF variables shown in Table 4 below. The netCDF header listing is provided in the Appendix in Section 8.1.

Value	netCDF History File Variable
1	Storm center latitude (degrees north)
2	Storm center longitude (degrees east)
3	daytime (seconds since 1970-01-01)
4	Raw T# unadjusted
5	Raw T# adjusted by Rule 8 rules
6	Final T# value (3-hour average)
7	Final CI# value
8	Wind Speed (m/s)
9	Eye region temperature (°C)
10	Mean cloud region temperature (°C) – movable annulus
11	Mean cloud region temperature (°C) – set region (old method)
12	“Coldest-warmest” cloud region temperature (°C)
13	Eye diameter, CDO size, or shear distance (in km)
14	Eye region temperature standard deviation value
15	Cloud region symmetry value
16	Satellite ID value
17	Eye region scene type (original value) 0=Clear Eye, 1=Pinhole Eye, 2=Large Eye, 3=No Eye
18	Cloud region scene type (original value) 0=CDO, 1=Embedded Center, 2=Irregular CDO, 3=Curved Band, 4=Shear
19	Eye region scene type (not used in HIE; no manual override functionality) Original value if user override performed, -1 if no override
20	Cloud region scene type (not used in HIE; no manual override functionality) Original value if user override performed, -1 if no override
21	Dvorak Weakening (Dvorak EIR Rule 9) flag 0=Off, 1=On; 2=On (initialized at >=T# 6.0
22	Dvorak Constraint Limits (Dvorak EIR Rule 8) flag 0’s=Shear Scene ,10’s=Eye Scenes, 20’s=/Curved Band/CDO/EmbC/IrrCDO Scenes 0=No Constraint, 1=T#4.0 0.5/6hr, 2=6hr rule, 3=12hr rule, 4=18hr rule, 5=24hr rule 6=None, 7=None, 8=0.2/hr (1st six hours only), 9=0.5/hr 30’s MW Adjustments : 30=12-hr Interp., 31=Initial, 32=On, 33=Hold, 34=Off
23	Latitude bias application flag 0=Off, 1=Merging, 2=On
24	Rapid dissipation flag (east pacific only) 0=Off, 1=On)
25	Land/ocean flag 1=Over Land, 2=Over Ocean
26	Eye region FFT value
27	Cloud region FFT value
28	Curved band analysis – BD curve gray scale value
29	Curved band analysis – convection curvature amount (out of 25)
30	Automated center determination flag (0=manual, 1=forecast interpolation, 4=spiral, 5=combo, 6=extrapolation
31	Distance from center to “coldest-warmest” temperature value
32	MSLP latitude bias adjustment value
33	Radius of maximum wind (in km)
34	External passive microwave (PMW) “eye score” value
35	Input 34 knot wind/gale radius value (for C/K/Z relationship; in km) (not used in HIE; C/K/Z not implemented)
36	Input environmental pressure value (for C/K/Z relationship, in mb) (not used in HIE; C/K/Z not implemented)

37	Storm number
38	Storm name
39	Basin ID value
40	Number of measurements in history file
41	CI# difference between current and previous record

Table 4: HIE history file parameter values.

3.3.2.2 Tropical Cyclone Forecast Product Formats

The HIE automated storm center location algorithm approximates TC position using three methods: polynomial interpolation of TCFC forecasts, linear extrapolation of the history file positions or with an advanced dual methodology examining the TC cloud top spiral features and circular feature (ring) search discussed in Section 3.4.2.1. Ancillary data, such as the operational forecast products described here, will be supplied to the HIE algorithm via the AIT Framework, which will obtain the data via the Ancillary Data Relay System (ADRS). Internal HIE software will obtain and read the input satellite data served by the AIT framework.

As a first guess, the forecast positions from TCFC forecast, with its local file name and directory location passed to the HIE algorithm via algorithm inputs , are read within the HIE algorithm to obtain the current position using a polynomial interpolation routine. This interpolation routine uses three forecast positions (current, 12 hour, and 24 hour positions) defined in the forecast file to interpolate the current position defined by the time of the image being examined. Two positions can be used in the case where the 24 hour forecast is not available, but three positions is generally preferred. If the image time falls within the time period defined within the forecast file positions/times, a valid interpolated forecast position will be used within the algorithm, otherwise an extrapolation of the previous history file positions will be used. The four types of acceptable input file formats are:

- ATCF Forecast Record file
- NHC Discussion/Warning files
  (WMO headers: “WTNT4? KNHC” or “WTPZ4? KNHC”)
- JTWC Tropical Cyclone Warning files
  (WMO header: “WTPN3? PGTW”)
- Generic Entry file

Depending on where the TC being investigated is located, some of the four formats may not be valid or available (ATCF are typically only available for storms monitored by US interests (NHC, CPHC) JTWC may not make available their ATCF files publically available. The order above is a general guideline for accessing the TCFC forecast information, but again these can vary depending on TC location, method of accessing the forecast information, and transmission by the specific TCFCs listed above. Examples of each valid forecast file format are provided below.

3.3.2.2.1 ATCF Forecast Record Format

The HIE will skip redundant forecast positions (for each critical radii report).

```
AL, 13, 2002092400, 03, OFCL,   0, 127N,  620W,  50, 1006, TS,  34, AAA,    0,    0,    0,    0,    0,    0,  0,  60,   0,    ,   0, BRJ, 275,  16,
AL, 13, 2002092400, 03, OFCL,   3, 127N,  627W,  50, 1006, TS,  34, NEQ,   90,   30,   50,   90,    0,    0,  0,  60,   0,    ,   0, BRJ, 275,  15,
AL, 13, 2002092400, 03, OFCL,   3, 127N,  627W,  50, 1006, TS,  50, NEQ,   20,    0,    0,   20,    0,    0,  0,  60,   0,    ,   0, BRJ, 275,  15,
AL, 13, 2002092400, 03, OFCL,  12, 138N,  648W,  55, 1006, TS,  34, NEQ,  100,   30,   50,  100,    0,    0,  0,  65,   0,    ,   0, BRJ, 300,  15,
AL, 13, 2002092400, 03, OFCL,  12, 138N,  648W,  55, 1006, TS,  50, NEQ,   20,    0,    0,   20,    0,    0,  0,  65,   0,    ,   0, BRJ, 300,  15,
AL, 13, 2002092400, 03, OFCL,  24, 150N,  671W,  60, 1006, TS,  34, NEQ,  100,   50,   50,  100,    0,    0,   0,  75,   0,    ,   0, BRJ, 300,  13,
AL, 13, 2002092400, 03, OFCL,  24, 150N,  671W,  60, 1006, TS,  50, NEQ,   30,   20,   20,   30,    0,    0,   0,  75,   0,    ,   0, BRJ, 300,  13,
AL, 13, 2002092400, 03, OFCL,  36, 161N,  690W,  70, 1006, TS,  34, NEQ,  100,   60,   60,  100,    0,    0,   0,  85,   0,    ,   0, BRJ, 300,  11,
AL, 13, 2002092400, 03, OFCL,  36, 161N,  690W,  70, 1006, TS,  50, NEQ,   50,   30,   30,   50,    0,    0,   0,  85,   0,    ,   0, BRJ, 300,  11,
AL, 13, 2002092400, 03, OFCL,  36, 161N,  690W,  70, 1006, TS,  64, NEQ,   30,   20,   20,   30,    0,    0,   0,  85,   0,    ,   0, BRJ, 300,  11,
AL, 13, 2002092400, 03, OFCL,  48, 170N,  705W,  75, 1006, TS,  34, NEQ,  120,   80,   80,  120,    0,    0,   0,  90,   0,    ,   0, BRJ, 300,   8,
AL, 13, 2002092400, 03, OFCL,  48, 170N,  705W,  75, 1006, TS,  50, NEQ,   50,   30,   30,   50,    0,    0,   0,  90,   0,    ,   0, BRJ, 300,   8,
AL, 13, 2002092400, 03, OFCL,  72, 190N,  730W,  70, 1006, TS,  34, NEQ,  100,   60,   60,  100,    0,    0,   0,  85,   0,    ,   0, BRJ, 310,   8,
AL, 13, 2002092400, 03, OFCL,  72, 190N,  730W,  70, 1006, TS,  50, NEQ,   50,   30,   30,   50,    0,    0,   0,  85,   0,    ,   0, BRJ, 310,   8,
```

3.3.2.2.2 NHC Discussion/Warning Record Format

These files can be obtained easily from various internet sources. The WMO header will change for each active storm (up to five active storms at a time) and each basin covered by the NHC and/or CPHC (Atlantic, East and Central Pacific).

```
WTNT42 KNHC 281454                              2005240 1454
TCDAT2
HURRICANE KATRINA DISCUSSION NUMBER  23
NWS TPC/NATIONAL HURRICANE CENTER MIAMI FL
11 AM EDT SUN AUG 28 2005
THE AIR FORCE HURRICANE HUNTERS JUST MEASURED A 166 KT FLIGHT LEVEL
WIND IN THE NORTHEAST EYEWALL...WHICH REQUIRES AN ADDITIONAL UPWARD
ADJUSTMENT OF THE CURRENT INTENSITY TO 150 KT.  A DROP IN THE EYE
GAVE A CENTRAL PRESSURE OF 907 MB.  KATRINA IS COMPARABLE IN
INTENSITY TO HURRICANE CAMILLE OF 1969...ONLY LARGER.  GPS
DROPSONDE DATA FROM THE NOAA G-IV MISSION EARLIER TODAY SHOWED
KATRINA'S INTENSE CYCLONIC CIRCULATION EXTENDING THROUGH THE 200 MB
LEVEL...WITH THE FLOW SPIRALING ANTICYLONICALLY OUTWARD IN A
WELL-DEVELOPED UPPER-LEVEL OUTFLOW PATTERN BEYOND A COUPLE HUNDRED
N MI FROM THE CENTER.  FLUCTUATIONS IN STRENGTH...DUE TO INTERNAL
STRUCTURAL CHANGES...ARE LIKELY PRIOR TO LANDFALL.  HURRICANES
RARELY SUSTAIN SUCH EXTREME WINDS FOR MUCH TIME.  HOWEVER WE SEE
NO OBVIOUS LARGE-SCALE EFFECTS TO CAUSE A SUBSTANTIAL WEAKENING THE
SYSTEM...AND IT IS EXPECTED THAT THE HURRICANE WILL BE OF CATEGORY
4 OR 5 INTENSITY WHEN IT REACHES THE COAST.

THERE IS NO CHANGE TO THE TRACK FORECAST.  KATRINA SHOULD GRADUALLY
TURN TOWARD THE NORTH...INTO A WEAKNESS IN THE SUBTROPICAL RIDGE
ASSOCIATED WITH A LARGE MID-LATITUDE CYCLONE OVER THE NORTHERN
UNITED STATES AND SOUTHERN CANADA.  THE OFFICIAL FORECAST TRACK IS
ABOUT IN THE MIDDLE OF THE DYNAMICAL GUIDANCE MODELS...WHICH ARE
RATHER TIGHTLY CLUSTERED.  RECALLING THAT THE AVERAGE NHC 24-HOUR
TRACK FORECAST ERROR IS ABOUT 80 N MI...THE ACTUAL LANDFALL POINT
COULD STILL BE ANYWHERE FROM SOUTHEASTERN LOUISIANA TO THE
MISSISSIPPI COAST.  ALSO...WE MUST CONTINUE TO STRESS THAT THE
HURRICANE IS NOT JUST A POINT ON THE MAP...BECAUSE DESTRUCTIVE
WINDS...TORRENTIAL RAINS...STORM SURGE...AND DANGEROUS WAVES EXTEND
WELL AWAY FROM THE EYE.  IT IS IMPOSSIBLE TO SPECIFY WHICH COUNTY
OR PARISH WILL EXPERIENCE THE WORST WEATHER.

THIS ADVISORY SHOWS AN ADDITIONAL EXPANSION OF THE WIND FIELD OVER
THE EASTERN SEMICIRCLE BASED ON AIRCRAFT AND SURFACE OBSERVATIONS.
HURRICANE FORCE WINDS ARE FORECAST TO SPREAD AT LEAST 150 N MI
INLAND ALONG THE PATH OF KATRINA.  CONSULT INLAND WARNINGS ISSUED
BY NATIONAL WEATHER SERVICE FORECAST OFFICES.

FORECAST POSITIONS AND MAX WINDS

INITIAL      28/1500Z 26.0N  88.1W   150 KT
 12HR VT     29/0000Z 27.2N  88.9W   145 KT
 24HR VT     29/1200Z 29.1N  89.6W   140 KT
 36HR VT     30/0000Z 31.4N  89.5W    85 KT...INLAND
 48HR VT     30/1200Z 34.5N  88.5W    45 KT...INLAND
 72HR VT     31/1200Z 40.0N  84.0W    30 KT...INLAND
 96HR VT     01/1200Z 45.0N  77.0W    25 KT...EXTRATROPICAL
120HR VT     02/1200Z 52.0N  69.0W    25 KT...EXTRATROPICAL
```

3.3.2.2.3 JTWC Tropical Cyclone Warning Record Format

As with NHC Bulletins, these forecasts can be readily obtained from the internet, and will have a separate WMO header for each storm and each basin covered by the Joint Typhoon Warning Center.

```
WTPN32 PGTW 291500                              2004242 1417
MSGID/GENADMIN/NAVPACMETOCCEN PEARL HARBOR HI/JTWC//
SUBJ/TROPICAL CYCLONE WARNING//
RMKS/
1. TYPHOON 19W (CHABA) WARNING NR 044
  02 ACTIVE TROPICAL CYCLONES IN NORTHWESTPAC
  MAX SUSTAINED WINDS BASED ON ONE-MINUTE AVERAGE
   ---
  WARNING POSITION:
  291200Z4 --- NEAR 29.4N5 130.0E4
    MOVEMENT PAST SIX HOURS - 325 DEGREES AT 07 KTS
    POSITION ACCURATE TO WITHIN 040 NM
    POSITION BASED ON CENTER LOCATED BY A COMBINATION OF
    SATELLITE AND RADAR
  PRESENT WIND DISTRIBUTION:
  MAX SUSTAINED WINDS - 095 KT, GUSTS 115 KT
  RADIUS OF 064 KT WINDS - 050 NM NORTHEAST QUADRANT
                         050 NM SOUTHEAST QUADRANT
                         050 NM SOUTHWEST QUADRANT
                         050 NM NORTHWEST QUADRANT
  RADIUS OF 050 KT WINDS - 080 NM NORTHEAST QUADRANT
                         080 NM SOUTHEAST QUADRANT
                         080 NM SOUTHWEST QUADRANT
                         080 NM NORTHWEST QUADRANT
  RADIUS OF 034 KT WINDS - 200 NM NORTHEAST QUADRANT
                             OVER WATER
                         180 NM SOUTHEAST QUADRANT
                         180 NM SOUTHWEST QUADRANT
                         180 NM NORTHWEST QUADRANT
  REPEAT POSIT: 29.4N5 130.0E4
   ---
  FORECASTS:
  12 HRS, VALID AT:
  300000Z3 --- 31.6N0 130.3E7
  MAX SUSTAINED WINDS - 080 KT, GUSTS 100 KT
  RADIUS OF 064 KT WINDS - 050 NM OVER WATER
  RADIUS OF 050 KT WINDS - 080 NM OVER WATER
  RADIUS OF 034 KT WINDS - 200 NM NORTHEAST QUADRANT
                             OVER WATER
                         180 NM SOUTHEAST QUADRANT
                             OVER WATER
                         180 NM SOUTHWEST QUADRANT
                         170 NM NORTHWEST QUADRANT
                             OVER WATER
  VECTOR TO 24 HR POSIT: 030 DEG/ 19 KTS
   ---
  24 HRS, VALID AT:
  301200Z6 --- 34.9N6 132.7E3
  MAX SUSTAINED WINDS - 060 KT, GUSTS 075 KT
       ---
REMARKS:
291500Z7 POSITION NEAR 30.0N3 130.1E5.
TYPHOON (TY) 19W (CHABA), LOCATED APPROXIMATELY 215 NM SOUTH
OF SASEBO, JAPAN, HAS TRACKED NORTHWESTWARD AT 07 KNOTS OVER
THE PAST 06 HOURS. THE WARNING POSITION IS BASED ON 291130Z6
ENHANCED IR SATELLITE IMAGERY.
```

3.3.2.2.4 Generic Format

Each line in the Generic Entry file contains either the initial/current position of the storm or a forecast position at 12-hours or 24-hours, and is formatted as follows:

dd mm yyyy tttt aa.a bbb.b

parameter	value	format
dd	date	two digit integer
mm	month	two digit integer
yyyy	year	four digit integer
tttt	UTC time	HHMM format : HH-hour MM-minute
aa.a	latitude	floating point (+/- : north/south)
bbb.b	longitude	floating point (+/- : west/east)

An example file for Hurricane Floyd using the 12 September 1999/03:00UTC forecast product:

```
12 09 1999 0300 22.7 64.5
12 09 1999 1200 22.9 66.0
13 09 1999 0000 23.2 68.3
```

The first line contains the "initial" position, with the second and third lines containing the 12-hour and 24-hour forecast positions. Use of a forecast file older than 24 hours from the initial time of the forecast file will result in a failure with the forecast interpolation routine since the image being examined will fall outside of the time range being read by the HIE algorithm (which only uses the initial position/time, the 12 hour forecast position, and the 24 hour position). If the time of the image being examined by the HIE falls within the current and 24 hour times defined in the forecast file being used, a valid forecast position will be determined. In addition, if the time of the image being examined falls before the initial time within the forecast file (as can happen if the TCFC releases the forecast file prior to the initial time within the forecast file), the interpolation routine will also fail since the time being interpolated to falls outside of the three positions defined in the forecast file. A failure to properly interpolate the forecast position will result in an extrapolation of the history file (if utilized, as described in Section 3.4.2.1), or termination of the HIE algorithm if an extrapolation cannot be performed.

3.3.2.3 Topography File Format

TC land interaction will be determined utilizing the high-resolution digital elevation map digilev_hires_le.map provided in the HIE package. The current file supplied with the HIE package uses an 8-km surface elevation map, adopted from the GEOCAT library, which was originally derived from the USGS GTOPO30 dataset available on the USGS website at:

http://edcdaac.usgs.gov/gtopo30/gtopo30.html.

The algorithm used to read this topography file was supplied by AIT programmers and converted from FORTRAN90 to C for insertion into the HIE algorithm.

To determine if a storm is over land, the algorithm will parse the topography file for the value located at the storm center position. If this position is a land value the land flag will be triggeredand no intensity estimate will be produced. The history file will still include an entry for any intensity estimate attempted while over land. Typically the automated HIE will continue to attempt intensity estimates while the storm is over land until either the official TCFC forecasts are discontinued or the TC reemerges over open ocean again and has not dissipated while over the land surface. The date and time(s), latitude, longitude, land flag, internal HIE satellite ID, and automated centering method values will contain their assigned values, however the values for the intensity estimates and the remaining parameters will be assigned a “missing value”. All values will contain zero for their missing values except the four temperature values (99.50), the radius of maximum wind value (-99.5), and microwave eye score value (-99.5). Calculation of both Final T# values and the CI# value will exclude all “over land” entries once the storm resumes movement over open water (non-land grid points).

3.3.2.4 Passive Microwave Eye Score File

The external passive microwave “eye score” algorithm, described in Section 3.4.2.12, will provide the ancillary date, time, and score value for the HIE when available. This information is currently scheduled to be derived and supplied by NOAA/NESDIS/SAB for the HIE algorithm via the AIT Framework, which will obtain the data via the Ancillary Data Relay System (ADRS). Internal HIE software will obtain and read the input satellite data served by the AIT framework. The score value should be a floating point value, while the date and time formats should be consistent with the corresponding data value (Value 1 and 2) formats in the HIE history file, described in Section 3.3.2.1.

The data files will be an eight-byte ASCII file containing the eye score value. The date and time of the MW overpass of the storm will be contained within the file name, which has the following format:

yyyymmddThhnnss_sat.dat
yyyy=year, mm=month, dd=day, hh=hour, nn=minute, ss=second, sat=satellite ID

3.3.2.5 Courtney/Knaff/Zehr Wind-Pressure Relationship Parameters

A new relationship between the TC wind and MSLP values has been implemented into the HIE using the methodology outlined in Courtney and Knaff (2009) and Knaff and Zehr (2007). This technique utilizes several environmental TC parameters to convert the T# value obtained by the HIE to wind speed (using the relationship shown in Table 6) to an estimate of MSLP.

The environmental parameters needed by the HIE to calculate the MSLP estimate are the environmental MSLP and the gale radius (the 34 knot wind radius) of the TC being examined. The environmental MSLP value can be obtained by determining the outermost closed isobar and adding 2 hPa, while the gale radius can be derived by averaging all available quadrant 34 knot wind radii estimate values. These values are obtained from the ATCF Best Track/Objective Aid/Wind Radii format files using the observational comprehensive archive (CARQ) record values. The ATCF homepage is provided in Section 1.4. The environmental MSLP/last closed isobar value is the "RADP" value, while the 34 knot radii values (value of 34 in the RAD column) will be stored in the RAD1 through RAD4 values.

The current equation for deriving the MSLP value is:

$$MSLP = 23.286 - (0.483 * Vstorm) - \left(\frac{Vstorm}{24.254}\right)^2 - (12.587 * S) - 0.483 * ABS(Lat)) + Penv$$

Where: Penv = Environmental MSLP (input from ATCF value)
Lat = Storm Center Position Latitude value
S = MAX(Vstorm500/Vstorm500C),0.4)
Vstorm = Vmax – (1.5*StormSpeed**0.63)
StormSpeed = 11.0 knots (constant value)
Vstorm500 = (R34/9.0) – 3.0
Vstorm500C = Vstorm*((66.785*Vstorm) + (1.0619*(ABS(Lat)-25.0)))/500.0)**A
A = 0.1147 + (0.0055 * Vstorm) – (0.001 * (ABS(Lat) – 25.0))
R34 = Gale Radius (average radius of 34 knot winds, input from ATCF value, if available
If R34 not available from ATCF it can be approximated by the equation
R34 = (0.354 * ROCI) + 13.3
ROCI = Radius of last closed Isobar (input from ATCF value)

3.4 Theoretical Description

3.4.1 Physics of the Problem

The HIE algorithm is based upon the simple premise that various cloud patterns and cloud top temperatures values/fields are related to the current tropical cyclone intensity. Tropical cloud regions that are less organized possess isolated and short-lived convective regions. As these regions become more organized, the convection becomes stronger and more widespread. The released latent heat from the convection supplies further energy to the convective region to increase the duration and vigor of the convection, leading to stronger wind speeds as the pressure within the organizing tropical system decreases related to the surrounding environmental pressure. The process continues to feed upon itself, with stronger convection releasing greater amounts of latent heat and a further reduction in the storm central pressure, which increased the geostrophic and ageostrophic wind speed wind around the storm.

As the convection continues to organize and intensify the satellite-observed cloud top temperature values decrease as the tops of the storms reach higher into the troposphere. Further increase in storm organization enhances the TC secondary circulation, leading to a more symmetrical convective appearance and the development of an eye feature at the center of the storm system. Various environmental factors, such as wind shear, sea surface temperature changes, and land/weather system interactions will limit the maximum intensity of the storm and/or lead to a decrease in the storm intensity through a disruption the convective processes driving the storm development.

Monitoring the changes in convective cloud patterns and cloud-top temperature values from satellite platforms over time can provide vital information about the organization and strength of the storm being observed. Relating these patterns and values to storm intensity is the objective of the HIE. Utilization of infrared imagery from geostationary satellites provides the ideal data to perform this function.

3.4.2 Mathematical Description

The designation of the current image scene type, the derivation of the hurricane intensity estimates, and the determination of the automated storm center position estimate all involve complex mathematical processes. These processes will be discussed in the following sections.

3.4.2.1 Automated Storm Center Determination

The HIE automated storm center determination process utilizes an interpolation of an official TCFC short-term track forecast as a first guess for the storm center position. Official forecasts can be obtained from various World Meteorological Organization (WMO) sanctioned Regional Specialized Meteorological Center (RSMC), such as the

NOAA National Hurricane Center (NHC) for the Atlantic Ocean, Caribbean Sea, and Eastern Pacific Ocean, the Central Pacific Hurricane Center (CPHC) for the Central Pacific Ocean (to dateline), or the Joint Typhoon Warning Center (JTWC) for all three basins. A list and examples of all RSMC valid forecast products used by the HIE is provided in Section 3.3.2.2. Forecasts are typically provided from NHC, CPHC, or JTWC every six hours for active TCs at or above Tropical Depression classification. These forecast positions are given every 12 or 24 hours, and the HIE normally uses the initial position and the first 12-hour forecast point for its interpolated values to the current analysis time.

The HIE will interpolate the provided TC position forecast to the current satellite image time using a polynomial interpolation of the storm forecast positions. The polynomial interpolation routine was derived from the “polint” program in Numerical Recipes in FORTRAN 77 – The Art of Scientific Computing and rewritten into C. The polynomial routine can work with two or more forecast positions (including the initial “zero hour” position and one forecast position), but it is desirable to have more than two points.

If the forecast interpolation fails for any reason, a linear extrapolation of storm positions from the previous 12 hours (stored in the history file) is attempted. A minimum of four history file positions must be used in the extrapolation process. If the interpolation and extrapolation routines both fail, the HIE algorithm will fail, the appropriate error code will be displayed, and the algorithm will end.

Once a valid interpolated forecast or extrapolated history file position is obtained, the Final T# intensity estimate (explained later in the document) of the record immediately proceeding the current analysis time in the history file is found. If this value is less than 3.5, the interpolated/extrapolated point is used as the final automated storm position. However, if this value is greater or equal to 3.5, a thorough analysis of the image is conducted to determine if a potentially better storm center location can be employed.

Two techniques are used in the auto-centering scheme (Wimmers and Velden, 2004). These methods are quite computationally since the routines require the satellite image being investigated to be remapped into a rectilinear projection before proceeding. Both processes are always performed during the auto-centering process.

The remapping to rectilinear projection process involves transforming the satellite data, most likely in the original satellite projection (but it could be in any other projection such as Mercator), into an equally spaced grid in latitude/longitude space. This results in a depiction of the tropical cyclone which is independent of the original view from the satellite, which is required by the auto-centering routines. Any reprojection software can be used for this step, but the resulting image (which is memory-based) is a 100x100 2-dimensional array of brightness temperatures centered on the tropical cyclone. The source and remapped images should have the same resolution.

The reprojection technique used by the HIE is as follows. The destination 2-D array is subdivided into 3x3 boxes. A navigation transformation is performed to convert from line/element to latitutde/longitude at each of the box corners. It then determines the line/element coordinates in the source image that corresponds to these same latitude/longitude values. Those corner values are used in a bi-linear interpolation scheme (which is much faster than a navigation transformation) to compute the line/element coordinate of each source image pixel it will use to fill the intermediate points in the box. This method ensures that each pixel in the destination array has an associated pixel from the source image.

For example, here is a schematic diagram of how it interpolates three values within a box of four points:

```
1,1  B  1,2

 A   C

2,1      2,2
```

Find the distance 'A' and 'B':

ZA=Z12-Z11
ZB=Z21-Z11

The value of 'C':
ZC=Z22-Z21+Z11-Z12

Compute new value IVAL, using the fractional distances XDEL, YDEL within the grid box. If XDEL=YDEL=0.5, then you are at the center of the box.

IVAL=NINT(Z11+ZA*XDEL+ZB*YDEL+ZC*XDEL*YDEL)

This is done separately for line and element values.

The routines that perform the bi-linear interpolation process were originally provided by Dave Santek of the Space Science and Engineering Center in Madison, WI, and were converted from the original FORTRAN code to C for the HIE algorithm. A much more detailed discussion of the process is provided in the Appendix, Section 8.5.

The first technique, the Spiral Centering (SC) routine, determines a center point in the image where there is a maximum alignment between the image cloud top temperature gradient and the 5-degree log spiral vector emanating from the center point within each different analysis region (at different points within the remapped rectilinear grid, as described below). The equation for the 5-degree log spiral is:

$$R(\Theta) = A * e^{(B * \Theta)}$$

$R(\Theta)$ = radial distance from origin (center point of SC analysis regions)
A = starting position distance of spiral from origin
B = angle between spiral tangent and radial line at point (R,Θ) (5° in this case)
Θ= angle from the x-axis

The log spiral must be converted to Cartesian coordinates in order to determine the vector at each point within the cloud top temperature grid field. If the origin of the analysis grid is center location located at (0,0), any point on the log spiral can be defined as:

$$X(\Theta) = R(\Theta) * \cos(\Theta)$$
$$Y(\Theta) = R(\Theta) * \sin(\Theta)$$

In the northern hemisphere, the X axis would increase positive moving east and the Y axis would increase positive moving north. In the southern hemisphere, the X axis would be the same, but the Y axis would increase positive moving south. This will assure the log spiral will spiral outwards in the correct direction for each hemisphere to match the cyclonic curvature of the storm rotation.

This analysis is a two-step procedure consisting of two analysis processes. The size and resolution of the different analysis grids are hardcoded within the HIE algorithm, with a fine-resolution analysis grid embedded within a course-resolution grid. The first step is a "course-resolution" analysis, where the temperature gradient and log spiral analysis vectors are calculated at individual grid points (the origin point $R(\Theta)$ in the log spiral equation) within a 1.75°-radius circular analysis region at a "course" resolution of 0.2°. The analysis is centered at the initial, interpolated forecast guess position provided from the TCFC forecast being used (or extrapolated history file position if the TCFC forecast is not available or valid for the image time being analyzed). Once an intermediate maximum alignment center position is determined using the cross-product analysis, a second "fine resolution" analysis is performed in a rectangular region surrounding the initial center position at a 0.1° "fine scale" resolution to find the final spiral center determined position.

The resolutions of the course and fine scale analysis grids were chosen in order to try and optimize the computation time of the SC routine versus the ability of the routine to capture the rotational features of the image being examined. Since the routine is processing the image over an image of size N x N at each point within the analysis region at different analysis resolutions, any increase in analysis resolution will result in an increase in processing for each pixel on the order of N^4. As computing power improves, an increase of the analysis resolutions can be considered (to eventually match the resolutions of the imagery being investigated), but currently the values utilized are optimized to allow for efficient and accurate processing.

The alignment is calculated by summing the cross-products of the image normalized temperature gradient vector with an overlaid spiral unit vector field (Figure 2). The normalized temperature gradient (NTG) is calculated using the following steps:

- Compute the normalized temperature gradient vector ($\overrightarrow{NTG}$):

- Compute the 2-D BT gradient field in units of K deg^{-1} ($\vec{\nabla} T_b$)
- Compute gradient magnitude: $GM = [(\vec{\nabla} T_b)^2_{lon} + (\vec{\nabla} T_b)^2_{lat}]^{0.5}$
- Compute gradient magnitude reduction factor: $GMR = \log{(1 + GM)}$
- Finally, $\overrightarrow{NTG} = GMR \cdot \vec{\nabla} T_b$

The equation for the cross product is:

$$\mathbf{a} \times \mathbf{b} = a*b*sin\Theta * \mathbf{n}$$

where:

a and b = magnitudes of vectors **a** and **b**
Θ = smaller angle between vectors **a** and **b**
n = unit vector perpendicular to the plane containing vectors **a** and **b**

As can be seen, the cross product value between the NTG vector and the spiral vector will be maximized where the angle between the vectors is 90°. The vectors **a** and **b** can be defined as either the NTG or the spiral vector field, but for this discussion, **a** is assigned to NTG vector and **b** is the spiral vector.

The cross product field is derived for both the coarse region and the fine-scale regions. Within the coarse analysis region, compute the cross product field of log spiral and normalized temperature gradient, iterating over center-points of the log spiral spaced by 0.2 degrees in two dimensions. Use the location of the maximum average cross product field as the second-stage center position. A smaller, finer resolution (0.1-degree) rectangular grid is centered on the second-stage center position determined in the course analysis. The final spiral center position is determined using the same cross product method as above. (The corresponding max cross product value is the Maximum Spiral Score or MSP).

The initial SC analysis MSP field normally takes shape as a “bull’s eye” pattern around the TC’s center of rotation. It is possible for this maximum alignment to occur along a curved convective band far from the storm center. An additional “distance penalty function” field (a weak “bull’s eye” pattern centered on the first guess forecast position) is added to the initial spiral center analysis field in order to discount such cases, after both the course and fine scale analysis has been performed, to guide the storm center determination process away from the edges of the spiral center analysis region. The highest distance penalty values are applied around the first guess position (innermost ring), and decreasing outward. The final “enhanced” spiral analysis field results.

This approach has the advantage of using the composition of the whole image instead of just the orientation of the spiral bands. Also, by using only the curvature of the image, the algorithm is not drawn to high-temperature “moats”, but rather uses their curved shape as a guide to the optimal center. The technique closely mimics the method used by experienced meteorologists to determine a center of rotation because it naturally assigns appropriate weights to the rotation-induced gradients of all sizes. However, the algorithm can be quite sensitive to the effects of upper-level shear in IR imagery, leading to center estimates driven by patterns in the asymmetric cirrus clouds rather than low-level clouds defining the center.

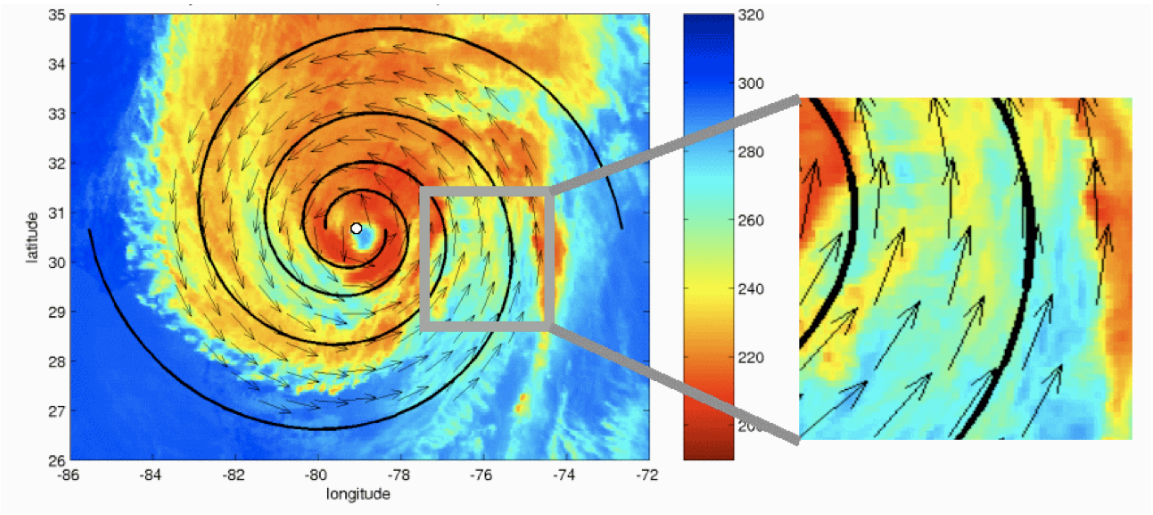

Figure 2: Example of a 5-degree log spiral field vector field. Black lines represent two spirals emanating from test center point (white dot). Arrows represent vector of log spiral at various points in analysis region.

Further calculation beyond the SC method is needed to resolve the exact center of rotation in storms which possess a well-defined eye (if one is present in the image being analyzed). Although finding the eye of a storm is a simple task for the meteorologist to perform subjectively, it is made objectively and computationally difficult by partial obscurations and nearby, oddly shaped high-temperature moats. To address this issue, the second technique is attempted at the completion of the SC process. The Eye Ring Fitting (RF) analysis method performs a search around the enhanced spiral analysis center point for the most intense gradients in a small ring-shaped area defining the storm eyewall. The dot product of the gradient field is summed for each point on each individual ring and center point, with the maximum value indicating the determined center point and TC eyewall size (Figure 3). The dot product equation is the following:

$$\mathbf{a} \cdot \mathbf{b} = |a| * |b| * \cos\Theta$$

where:
Θ = angle between a and b vectors

In the RF routine, **a** would be the temperature gradient vector at a specific point on the ring and **b** would be the “position” unit vector from the center point to the ring at the point being analyzed.. When the vectors are in the same direction, the angle will be near zero, maximizing the product of the vector magnitudes. This process is performed over a circle of radius 0.75° from the center position corresponding to the spiral analysis center point determined in the SC routine. The RF analysis iterates across all points inside the

0.75° radius. Those points then serve as the center points used in an inner loop that iterates across a range of smaller ring radii (between 0.06° and 0.40°). The resulting field is the final RF analysis score field. Note, no RF position is derived from the RF routine, only an analysis field, which is then combined with the SC analysis field described above using the method described in the following discussion.

It must be further noted that a "Moat Mask" identification routine is performed prior to the start of the RF routine to try and identify "false eye" regions that are not circular in structure. The Moat Mask routine derives a Connected Component Label mask field that inventories each contiguous patch of pixels below the threshold value of 237K within the RF analysis region. The moat regions are defined by determining a "feature length" value for each contiguous patch within the analysis region. The feature length (FL) is defined as:

$$\text{FL} = \sqrt{((MaxLAT - MinLAT)^2 + \frac{((MaxLON - MinLON)}{cos(\frac{AvgLAT}{180*\pi})})^2}$$

Any region with a FL value > 0.25° is considered a moat region and excluded excluded from being analyzed by the RF routine, with the dot score value being assigned a 0 for all points in the moat.

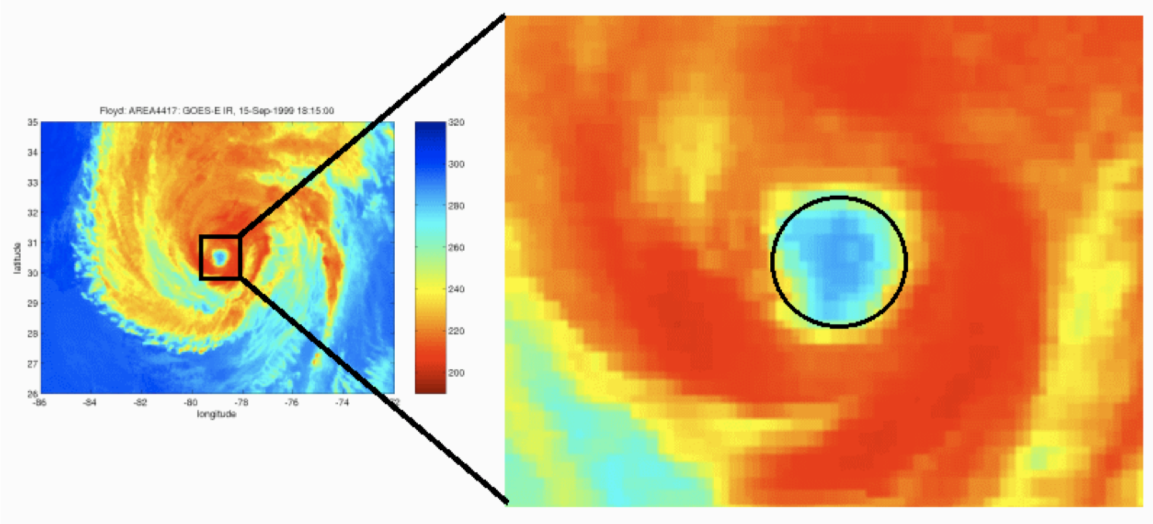

Figure 3: Example of eye ring analysis (black circle) which best matches temperature gradients of the eyewall feature in the tropical cyclone.

Once both analyses techniques have successfully completed, their derived analysis fields and corresponding confidence factors are combined to determine the "best" storm position for the scene being analyzed. The confidence factors score derivation process is performed in two steps. For each point an Enhanced Spiral Part (ESP) confidence score is derived by determining the minimum value of the sum of two weighted spiral values, as shown in the following equation:

ESP score = Distance Penalty (DP) + Spiral Part score (SP)

DP = constant * distance of point from First Guess position
SP = 10 * (spiral score at point – maximum spiral score)

Where "spiral score at point" is equal to the spiral score (from the SC routine process described previously) at each point in the analysis and "region maximum spiral score" is the maximum spiral score value within the analysis region. Basically this value is just the difference between the spiral score value at each individual point in the analysis region from the maximum spiral score value within the entire region.

If the position of the maximum ESP value (least negative) is outside of a set radii distance threshold (the maximum allowable displacement value of 1.15°) from the first guess position (forecast interpolation position), the automated storm position will result in the interpolated first guess position from the TCFC forecast or extrapolated position from the HIE history file (ID value of 5 or 6, respectively, for parameter 30 in the history file listed in Section 3.3.2.1 and Diagnostic Output in Section 3.4.3.1.3

If the position of the maximum ESP value (least negative) is within the set radius threshold of 1.15°, however, a second "distance weighted" Enhanced Spiral Part (ESPd) confidence score will be derived, adding the "Distance Bonus (DB)" value to the initial ESP value. The DB value is a constant value (equal to 4.5) which will further weight all ESP score values within 0.25° of the derived Spiral Center position derived previously. This is done to further weight the scores in the calculation described next towards the previously calculated Spiral Center position.

A third confidence factor will be also be derived utilizing the RF routine analysis values. This value is simply a sum of the ESPd value and the ring fit analysis score value at the position being analyzed. This value is called the Combined Confidence (CC) factor value. If the maximum CC factor value is greater than the empirically determined value of 15.0, the position of the maximum CC factor will be used as the final automatically-derived storm center position. This will result in the "Combination Ring/Spiral" automated storm centering method type being used (ID value of 5 for parameter 32 in the history file listed in Section 3.3.2.1) and identified as "COMBO" in the History File Listing. A maximum CC factor is less than or equal to 15.0 will result in the "Spiral Centering" automated storm centering method type being used (ID value of 4 for parameter 30 in the history file listed in Section 3.3.2.1 and in the Diagnostic Output in Section 3.4.3.1.3) and identified as "SPRL" in the History File Listing.

An example of an actual TC storm center selection process is shown in Figure 4. If the previous Final T# intensity was between 3.5 and 4.5, the Spiral Centering and Eye Ring Analysis positions will be used if three or more eye or embedded center scene types have been identified for previous intensity estimates prior to the current analysis date/time. This will ensure that the storm organization is sufficient for the different techniques to work properly and correctly estimate the storm center location. If less than three eye/embedded center scenes have been found, the forecast interpolation position will only be utilized. Once the Final T# value has exceeded 4.5, both auto-centering techniques will be attempted.

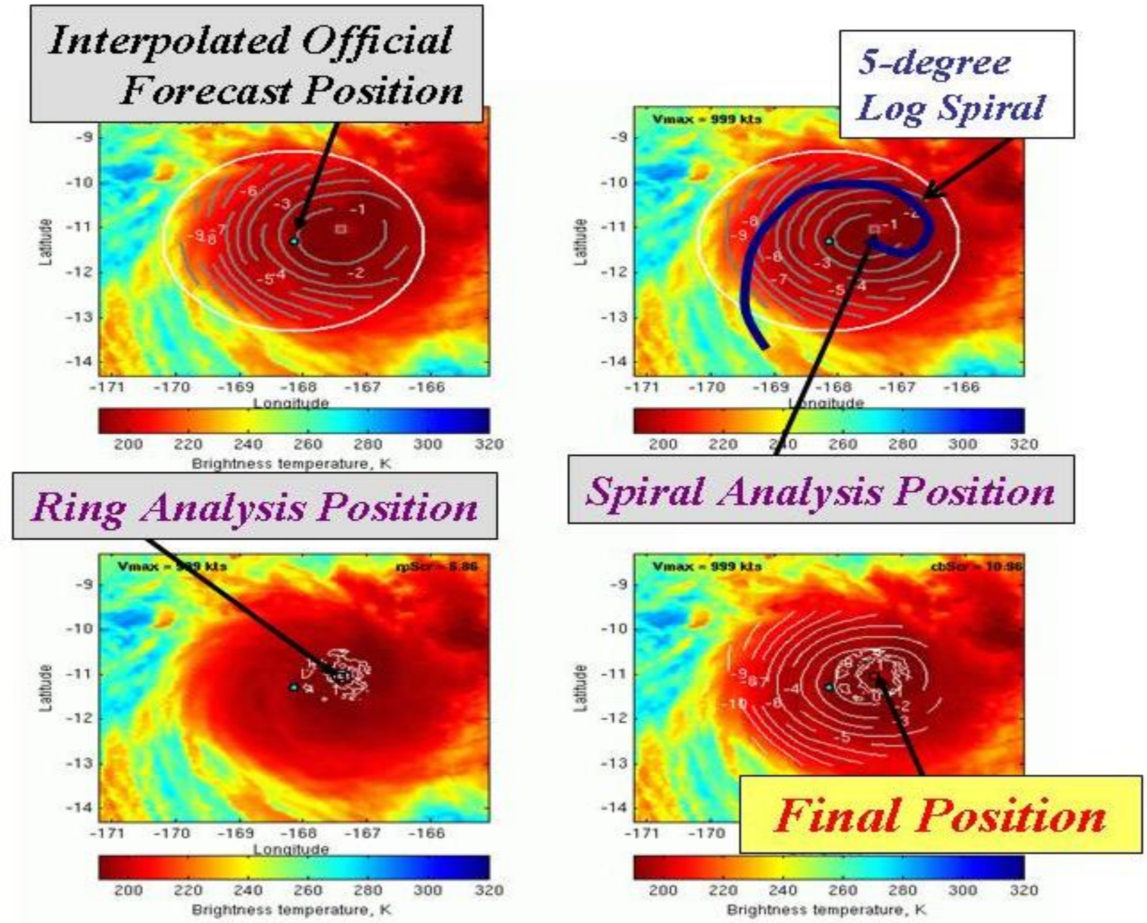

Figure 4: Four-panel image showing various steps, and resultant analysis for each step, in the automated storm centering routine. Upper two images show the initial and enhanced Spiral Centering (SC) contour analyses fields. The white oval represents the computation boundary of the SC analysis. Upper left is initial spiral analysis gradient field, with the enhanced SC in upper right (an example 5-degree log spiral is shown with blue line). Lower left shows contour field associated with Ring Fitting (RF) analysis. Lower right is combined SC and RF analysis contour field. Initial forecast position (blue dot) and subsequent intermediate and final center positions (red dots) are indicated.

3.4.2.2 Distance Determination

All distances calculated in the HIE are great circle distances. Specifically, the distance calculation is based upon the great circle chord length calculation for the angle and distance as presented in the following website:

http://en.wikipedia.org/wiki/Great-circle_distance.

Distances are provided in units of kilometers.

3.4.2.3 Scene Type Determination

The HIE scene type determination process flowchart is shown in Figure 5.

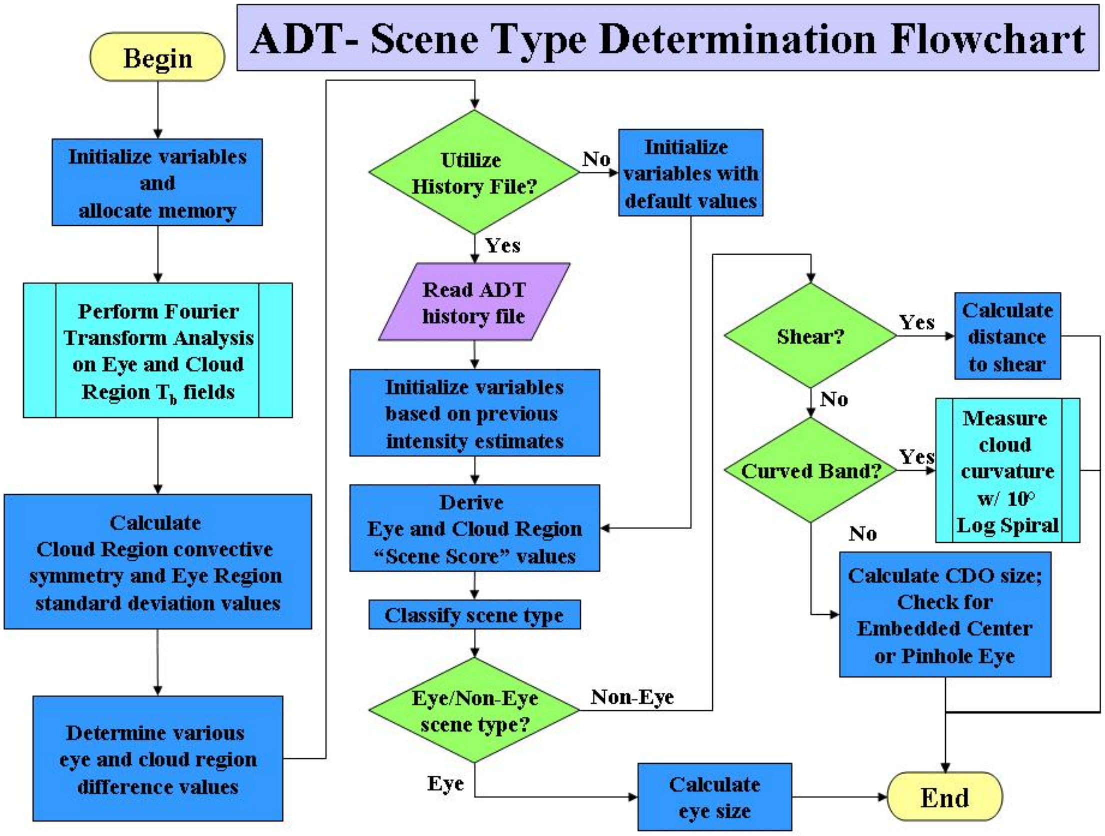
ADT- Scene Type Determination Flowchart
Begin
Initialize variables and allocate memory
Perform Fourier Transform Analysis on Eye and Cloud Region T_b fields
Calculate Cloud Region convective symmetry and Eye Region standard deviation values
Determine various eye and cloud region difference values
Utilize History File?
No
Initialize variables with default values
Yes
Read ADT history file
Initialize variables based on previous intensity estimates
Derive Eye and Cloud Region "Scene Score" values
Classify scene type
Eye/Non-Eye scene type?
Non-Eye
Eye
Calculate eye size
End
Shear?
Yes
Calculate distance to shear
No
Curved Band?
Yes
Measure cloud curvature w/ 10° Log Spiral
No
Calculate CDO size; Check for Embedded Center or Pinhole Eye

Figure 5: Scene Type determination flowchart.

3.4.2.3.1 Eye and Cloud Region Scene Score Determination

The original "decision tree" methodology of scene type determination patterned after the subjective Dvorak Technique has been replaced with a "scene score" determination scheme. This methodology helps alleviate the occasional rapid and unrealistic scene type variability in the HIE output between consecutive image analyses by replacing the "branches" in the decision tree methodology with natural threshold divisions (scores).

The scene score is a sum of selected environmental parameters measured from the satellite imagery for both the eye (when an eye is present, 0-24 km from the determined center position) and cloud region (24-136km from the center position) of the TC. These scores are used to objectively determine the storm scene type. Each parameter in the scene score derivation was originally a "branch" in the initial decision-tree methodology and is weighted by an empirically-determined value, shown in the equations listed below, chosen to closely mirror the influence of the parameter in the original determination methodology.

The eye region scene score is derived with the following equation:

Eye score = $FFT_{Eye} + BD_{Eye} + PrevEye + \Delta T_{Eye\text{-}Cloud} + Adj_{T\text{-}12hr}$

FFT_{Eye} : Number of harmonics in eye region Fast Fourier Transform
1.0-((EyeHarmonicValue – 2)*0.1)

BD_{Eye} : BD-enhancement category of eye region temperature
-1.0*(EyeBDCat*0.5)

PrevEye : Adjustment for previous record eye scene
+0.25

$\Delta T_{Eye\text{-}Cloud}$: Difference between eye and cloud region BD-enh. categories
(0.25*(CloudBDCat-EyeBDCat))+
(0.50*(CloudCWBDCat-EyeBDCat)

$Adj_{T\text{-}12hr}$: Adjustment for 12-hour previous record T# strength
MAX(-1.0,T#Value$_{T\text{-}12hrs}$ – 4.5)

The cloud region scene score is derived with the following equation:

Cloud score = $BD_{CloudCW} + BD_{Cloud} + FFT_{Cloud} + PrevScene + Adj_{T\text{-}12hr}$

$BD_{CloudCW}$: BD-enh. category of "coldest-warmest" cloud region temp.
CloudCWBDCat*0.25

BD_{Cloud} : BD-enhancement category of eye region temperature
CloudBDCat*0.25

FFT_{Cloud} : Adj. based on harmonics in cloud region Fast Fourier Trans.
If(CloudHarmonicValue<=2)value=MIN(1.5,BD$_{CloudCW}$*0.25)

PrevScene : Adjustment for previous record cloud scene
-0.50 if curved band or shear scene

$Adj_{T\text{-}12hr}$: Adjustment for 12-hour previous record T# strength
MIN(1.0,T#Value$_{T\text{-}12hrs}$ – 2.5)

More specific definitions and measurement details of utilized temperature categories and values, such as the "coldest-warmest" cloud region temperature value and BD enhancement category, are discussed here and in the following sections of this document.

- EyeHarmonicValue: Count of the number of times the harmonic values in the Fast Fourier Transform analysis increases over the previous value. Each harmonic value corresponds to a single cosine wave, at different frequencies, used to approximate the temperature histogram for the image region being investigated. More changes in the values indicate more harmonics and a less uniform temperature scene in the eye region.
- CloudHarmonicValue: Same as EyeHarmonicValue, but with cloud region.
- EyeBDCat: Gray scale category within the BD-curve enhancement of the eye region temperature value. The BD-curve enhancement is shown in Figure 7, with Table 5 listing the temperature breakdown values for each gray scale range..
- CloudBDCat: Same as EyeBDCat, but with cloud region temperature value.
- CloudCWBDCat: Same as EyeBDCat, but with cloud region "coldest warmest" temperature value.
- T#Value$_{T\text{-}12hrs}$: Final T# value at time 12 hours previous of current time.

The Fast Fourier Transform (FFT) analysis is a well-established and documented mathematical function to determine the variability of a "signal" over its range of values. In this case, the signal is the temperature histogram over a set temperature range. The FFT attempts to approximate this signal using a combination of a number of cosine waves with different frequency. The FFT algorithm used in the HIE is derived from code obtained online from Electronics Letters on January 5, 1984, which utilizes a Duhamel-Hollman split-radix difference FFT analysis. The code is supplied in the Appendix, section 8.6. A histogram with containing a large number of values in a small number of bins can be estimated with a low frequency cosine wave, while more chaotic histograms with more temperature value bins containing data can be approximated by the combination of several cosine waves at different frequencies.

The number of harmonic values is determined by parsing the output histogram array of the FFT computation and examining the bin values. There will be 32 FFT magnitude bins output from the FFT subroutine. To determine a "harmonic", the middle bin value is compared to the two surrounding values… if it is greater than the two surrounding bins it is counted as a harmonic and added to the output counter. So, starting with bin number 1, it is compared to bin 0 and bin 2. If it is the largest of the three bins, it is a harmonic. The process continues through all the bins where three comparisons are available… basically from bin 1 to bin 30.

Additional environmental variables are used in several specific checks to further refine eye and cloud scene types after the individual scene score values have been derived, such as the cloud symmetry value (discussed in Section 3.4.2.4.2) to help determine an 'Irregular CDO' scene, or eye radius size to define a' Large Eye' scene.

3.4.2.3.2 Eye and Cloud Region Scene Types

The three eye region scene types and five cloud region scene types are described below:

Eye Region Scene Type	Description
EYE	Any eye type (clear, ragged, and obscured)
PINHOLE	Very small eye/pronounced warm spot
LARGE	Clear, well-defined eye with radius >= 38 km

Cloud Region Scene Type	Description
UNIFORM CDO	Overcast cloud region with uniform temperature structure
EMBEDDED CENTER	Arc of convection within central overcast cloud region
IRREGULAR CDO	Cloud region over storm center, but large shift in coverage
CURVED BAND	Curved cloud region surrounding circulation center
SHEAR	Displaced convection and exposed circulation center

Parameters 17 and 18 in the netCDF history file, shown in Table 4, contain the eye and cloud scene ID values which correspond to the scene types listed above. Both parameters will always be determined, but the cloud scene type ID value is only considered when the eye scene ID value is equal to “3”. If the eye scene ID value is “3”, the cloud scene ID values will have values of “0” (for CDO), “1” (for Embedded Center), “2” (for Irregular CDO), “3” (for Curved Band), or “4” (for Shear). If the eye scene ID value is less than 3, the scene type is one of the eye scene types, and the cloud scene ID is ignored. An eye scene of “0” indicates and Eye (clear, ragged, or obscured), while eye scene ID values of “1” and “2” indicate a Pinhole Eye or Large Eye, respectively.

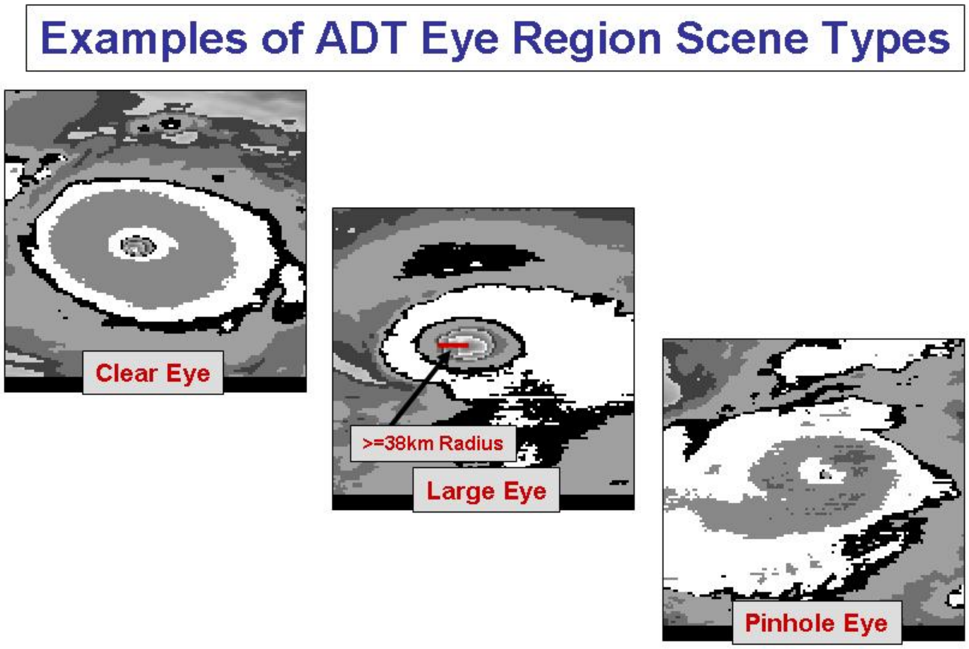

Figure 6: Examples of three HIE Eye Region scene types.

Examples of the three eye scene types are shown in Figure 6. Eye size is derived by measuring the distance across the eye between BD-curve "Dark Gray" temperature range edges (Figure 7). Table 5 lists the temperature ranges and the corresponding "gray scale" color values. Distances are measured at 90° angles outward from the auto-selected center location and are averaged to give an approximate eye size.

Temperature Range	Gray Scale Category/Comment	Value
> +9°C	No enhancement – Low clouds	0
+8.99°C to -30.0°C	Off White – Cirrus Outflow Pattern	1
-30.01°C to -42.0°C	Dark Gray	2
-42.01°C to -54.0°C	Medium Gray	3
-54.01°C to -64.0°C	Light Gray	4
-64.01°C to -70.0°C	Black	5
-70.01°C to -76.0°C	White	6
-76.01°C to -80.0°C	Top Medium Gray	7
< -80.01°C	Top Dark Gray	8

Table 5: BD-curve enhancement temperature and corresponding gray scale values.

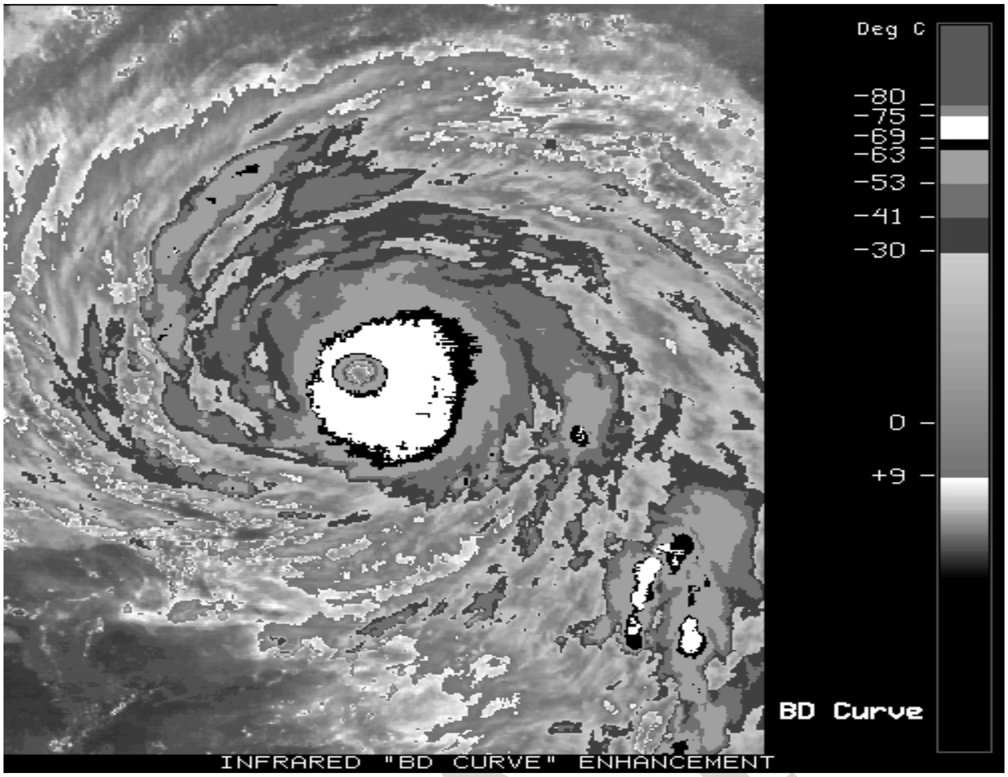

Figure 7: BD-curve Enhancement displayed on a longwave window channel infrared image.

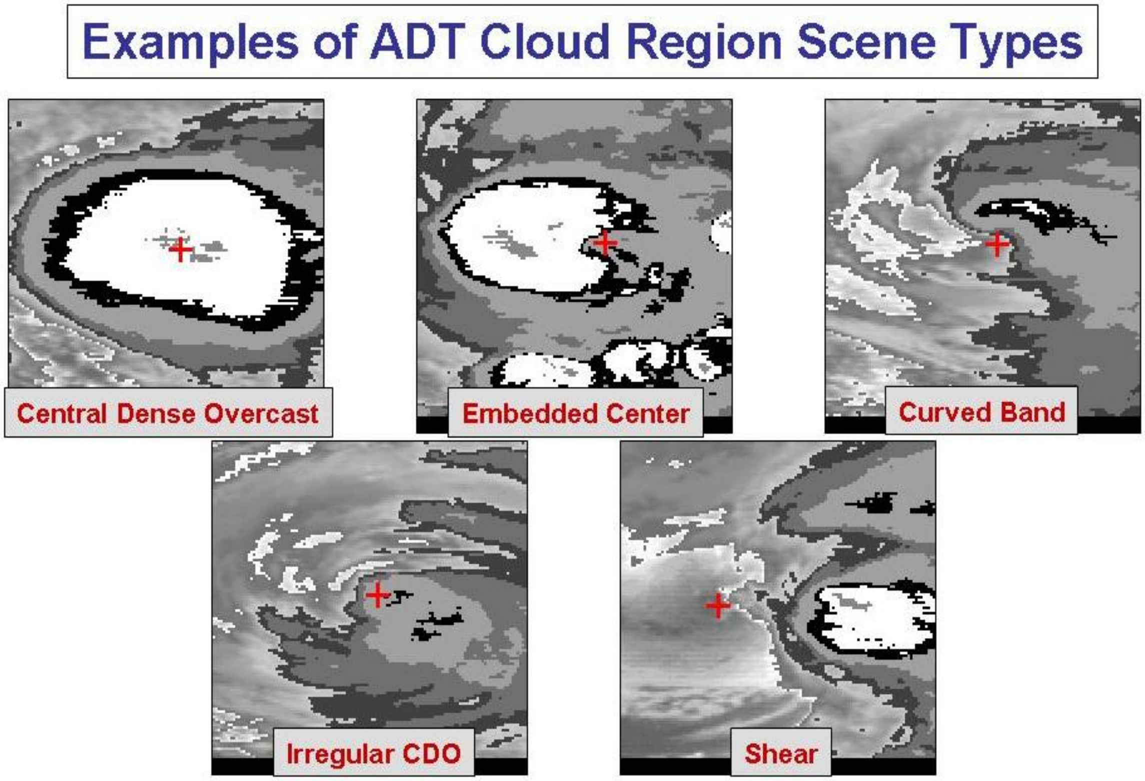

Figure 8: Examples of the five HIE Cloud Region scene types.

Examples of the five cloud region scene types are shown in Figure 8. If the cloud scene is determined to be a "Curved Band" type scene, a 10° Log Spiral analysis will be performed to determine the curvature extent of the convective cloud region around the selected center position (described in greater detail in Section 3.4.2.4.3). The amount of curvature depends on the BD-Curve Gray Scale (Table 5) being used. The algorithm will first target the "Light Gray" gray scale range to define the convective region of the curved band area. If this temperature region is devoid of significant activity or completely surrounds the selected storm center position in this gray scale range, analysis of warmer/colder ranges BD-curve temperature levels will be performed. Once a Curved Band gray scale and curvature amount at the provided storm center location is determined, an additional search will be performed to look for another possible storm center location that yields a greater amount of curvature in the cloud region, and thus a higher intensity estimate. This maximum curvature search value is supplied only as a reference to the HIE user and is not utilized in any way by the HIE algorithm or stored in the history file for future use/reference. It is supplied at the request of HIE users to help assess the accuracy of the curved band intensity calculation and to aid TCFC forecasters in the TC storm center determination process for possible storm center location adjustment. The location and associated amount of maximum cloud curvature can vary greatly from image to image, so interpretation of its value is left to the user to assess.

Once the scene type has been determined the intensity estimate for the derived scene type will be calculated using various environmental values such as the eye and surrounding cloud top temperatures, convective symmetry, eye and cloud region sizes, and other measured or derived values stored in the history file using various methods described in the following sections.

3.4.2.4 Eye and Surrounding Cloud Region Temperature Determination

Proper determination of the storm center position is paramount for an accurate storm intensity estimate since the eye and cloud region temperature value calculations are heavily dependent on this location.

Determination of the eye region temperature is relatively straightforward. This temperature is assigned the value of the warmest pixel within a 24 km radius from the user or automated storm center location. Since an incorrect eye temperature can lead to an incorrect intensity estimate, proper selection of the storm center location is essential. Note: The 'eye' terminology can be misleading, as this temperature is calculated even if a storm eye is not present. It basically represents a center, or core temperature, and is derived and utilized by the HIE even in non-eye scenes/situations.

The calculations of the two surrounding cloud region temperature values are a bit more complicated than for the eye temperature. The first cloud region temperature value computed uses the storm center location and lies between an annulus of 24 and 136 km from the center location. Pixel-width rings within this annulus are then analyzed (e.g., for a 4 km resolution infrared image, there will be (136 - 24)/4 = 28 rings). On each ring the warmest brightness temperature is found and stored. The coldest of these ‘warmest’ values is then selected, and referred to as the “coldest-warmest” temperature. It is used only in the determination process for the cloud region scene type. The ring number (CWRN) distance (in km from the storm center location) on which the “coldest-warmest” temperature value is located is used in the determination of the final “mean” cloud region temperature discussed below. For more information about this value, see Velden et al. (1998) and Zehr (1989).

The second temperature value computed is the “mean” cloud region temperature, and is the actual value that is assigned as the HIE’s final cloud region temperature used in subsequent intensity determinations. First, an 80km wide annulus region is determined around the storm center location, with the annulus midpoint centered on the CWRN distance explained above. Typically the minimum inner/maximum outer radius of the annulus will be the CWRN distance minus/plus 40 km. The minimum inner radius, however, is restricted to 24 km from the storm center location. There is no corresponding maximum limit on the outer radius distance. For example, if the CWRN distance is 48 km, the inner distance, if centered upon this point, would be at 8km. Since the minimum annulus radius cannot be less than the minimum value of 24 km (not 8 km), and the annulus will always be 80km wide, the outer distance will be 104km. Once the annulus is defined, it is divided into 24 arc regions (15° each) and the average temperature of each arc is calculated. The mean of the 24 average arc temperature values is computed and assigned as the final HIE cloud region temperature value.

The cloud region “coldest-warmest” (CloudCW) temperature value is a “historical” temperature value that mimics the cloud temperature value utilized in the original Dvorak EIR technique. The CloudCW temperature is found by looking at 4km thick rings in the cloud top region between 24km and 136km from the center, so in this case there will be 28 rings. Each pixel is assigned to an associated ring by determining its distance from the center position (using the latitude and longitude position associated with each pixel and the storm center location). Pixels between 24km (inclusive) and 28km (exclusive) will be part of ring #1, between 28km (inclusive) and 32km (exclusive) will be part of ring #2, and so forth. Once all of the pixels have been assigned, the warmest pixel on each ring is determined. The coldest of these warmest ring pixels is then used as the “coldest warmest” temperature value (with the corresponding minimum distance of the ring being determined and stored as the ring distance value).

The CloudCW value was utilized in older versions of the algorithm as the final cloud region temperature value, but it was replaced with the “mean” temperature value after a close examination of both values found that the CloudCW temperature value varied significantly more than the mean value. The intensity estimates computed using the coldest-warmest temperature values were also found to be inferior to the intensity estimates using the mean temperature values, thus the change was implemented. The CloudCW value is still utilized within the HIE scene selection process as described above, so the calculation of the value still remains within the HIE algorithm and the derived parameters are saved in the HIE history files.

3.4.2.5 Intensity Estimation Processing

The HIE hurricane intensity estimation processing scheme involves several steps, as shown in Figure 9. Many of the specific elements involved in the intensity estimation derivation process are discussed in detail in the following sections. Intensity constraint rules are discussed in detail in Section 3.4.2.6.

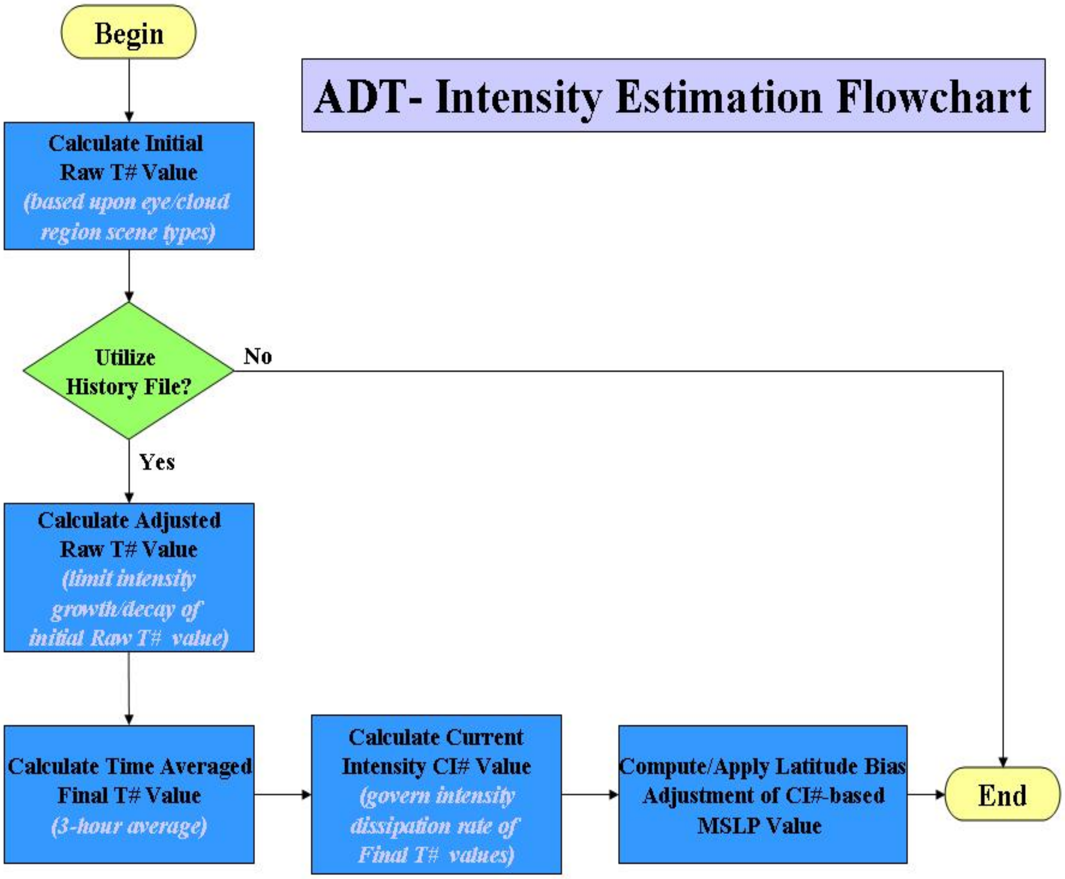
ADT- Intensity Estimation Flowchart
Begin
Calculate Initial Raw T# Value
(based upon eye/cloud region scene types)
Utilize History File?
No
Yes
Calculate Adjusted Raw T# Value
(limit intensity growth/decay of initial Raw T# value)
Calculate Time Averaged Final T# Value
(3-hour average)
Calculate Current Intensity CI# Value
(govern intensity dissipation rate of Final T# values)
Compute/Apply Latitude Bias Adjustment of CI#-based MSLP Value
End

Figure 9: HIE Intensity Estimation flowchart.

3.4.2.5.1 Intensity Estimate Derivation Methods

As discussed in the Algorithm Overview in Section 3.1, the HIE is a derivative of the ADT algorithm, which itself is originally based upon the operationally-utilized Subjective Dvorak Enhanced Infrared ‘EIR’ Technique (SDT). Several deviations from the original SDT have been included into the ADT in order to improve upon the initial implementation of logic from the SDT into the ADT. The primary advancement involves the utilization of linear regression equations relating various cloud parameters to TC intensity. These equations were implemented for all eye scenes and several of the cloud scenes described in Section 3.4.2.4.2. The remaining scene types not utilizing regression-based methods have also been improved to better correlate their specific cloud parameter measurements to TC intensity.

3.4.2.5.2 Regression-based Intensity Estimates

As mentioned previously, a regression-based analysis scheme was implemented in order to better correlate HIE eye and cloud region temperature measurement values with observed intensity. Measured values such as maximum eye region and minimum cloud region temperatures have been used historically in the SDT and early versions of the HIE to estimate TC intensity in all eye scene types as well as the CDO (and Irregular CDO) and Embedded Center cloud scene types. These parameters worked well in most cases, but additional methods were explored to better assess the TC intensity. Using linear regression analysis, additional parameters were found to aid significantly in the process of estimating TC intensity using infrared window imagery. A schematic representation of the regression-based parameters used in the HIE is provided in Figure 10.

The cloud symmetry value measures the temperature difference between opposing sides of the storm to assess the structure of the convection surrounding the eyewall region of the TC. Storms possessing a more symmetric convective structure (smaller temperature differences between opposing sections of the storm eyewall cloud region) are typically more intense than storms with asymmetric convection. This parameter is actually indirectly included in the SDT EIR technique via measurements of the width of the various BD-enhancement gray shades surrounding the storm center position as well in the “climatological” pattern-T# comparison diagrams. The cloud symmetry value is obtained by determining the average cloud top temperature value within the 24 sectors defined earlier (15-degree arcs, 80-km wide annulus from storm center with distance from eye pre-determined by warmest-coldest ring value). Opposing sector values (180-degree rotation) are differenced and summed around the storm center. These twelve values are then averaged to determine the final cloud symmetry value.

Another important parameter is the size of the central dense overcast, or CDO scene type, region in a non-eye situation. This parameter mimics the SDT which also makes use of a CDO size parameter to help assess the current strength of the storm. The impact of this value itself is not large, but it was found to be a significant contributor to estimating intensity in these situations when combined with the other parameters. The value is derived by simply averaging the diameter measurements of the storm CDO along four axes centered on the storm center position. The CDO edge is defined as the boundary between the Light Gray and Medium Gray shades in the BD-enhancement shown in Figure 7 and listed in Table 5.

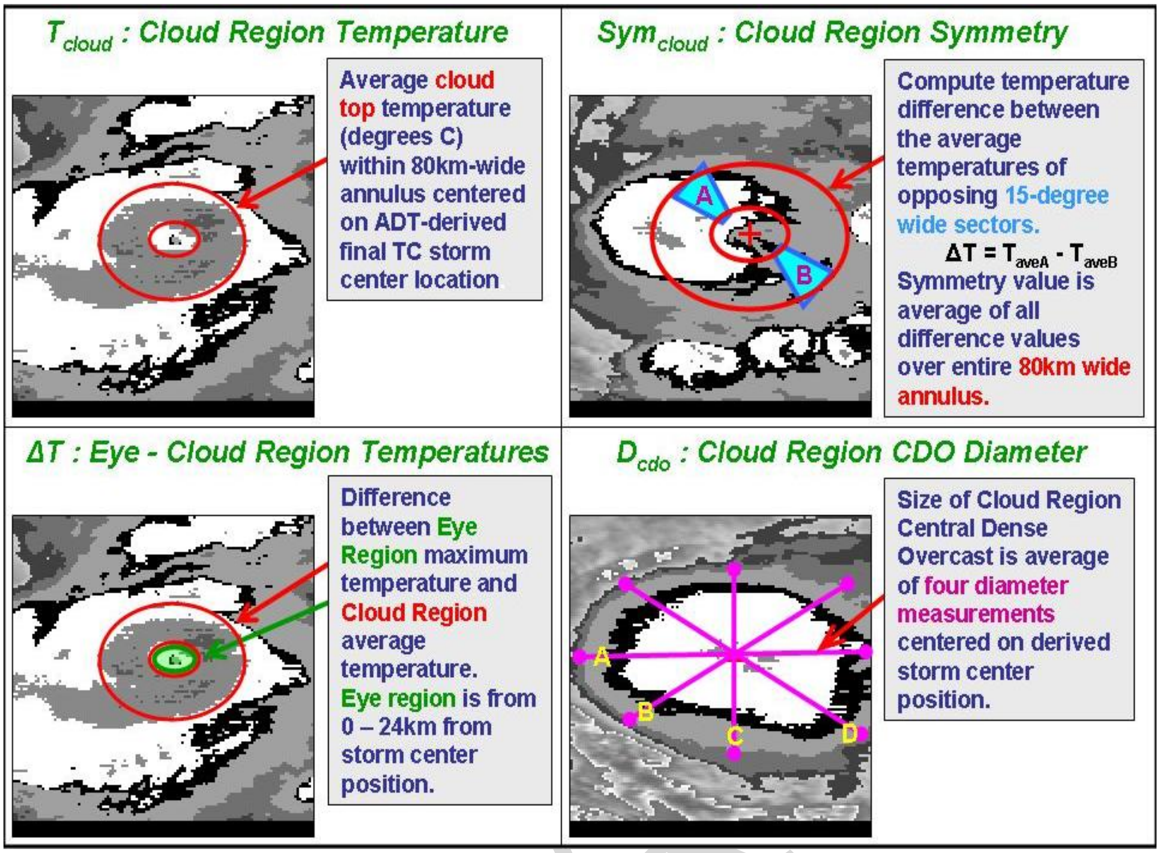
T_cloud : Cloud Region Temperature
Average cloud top temperature (degrees C) within 80km-wide annulus centered on ADT-derived final TC storm center location
Sym_cloud : Cloud Region Symmetry
Compute temperature difference between the average temperatures of opposing 15-degree wide sectors.
ΔT = T_aveA - T_aveB
Symmetry value is average of all difference values over entire 80km wide annulus.
A
B
ΔT : Eye - Cloud Region Temperatures
Difference between Eye Region maximum temperature and Cloud Region average temperature.
Eye region is from 0 – 24km from storm center position.
D_cdo : Cloud Region CDO Diameter
Size of Cloud Region Central Dense Overcast is average of four diameter measurements centered on derived storm center position.
A
B
C
D

Figure 10: Schematic representation of the four regression equation parameters.

Two separate regression equations are used in the HIE; one equation for the eye scene types and one equation for the cloud scene types. The equations yield intensity estimates in terms of T# values which can be converted to MSLP or Maximum Wind Speed (as discussed in Section 3.4.2.5.:

Eye Scenes: $\text{Intensity} = 1.10 - 0.070{*}T_{\text{cloud}} + 0.011{*}\Delta T - 0.015{*}\text{Sym}_{\text{cloud}}$

Cloud Scenes: $\text{Intensity} = 2.60 - 0.020{*}T_{\text{cloud}} + 0.002{*}D_{\text{cdo}} - 0.030{*}\text{Sym}_{\text{cloud}}$

T_{cloud}	: Cloud region average temperature (deg C)
ΔT	: Eye (max) – cloud (ave) region temperatures (deg C)
$\text{Sym}_{\text{cloud}}$	: Cloud region temperature symmetry (deg C)
D_{cdo}	: Cloud region CDO size (km)

In actuality, the base number (y-intercept: 1.1 or 2.6 in the eye/cloud equations) and the T_{cloud} parameters have been replaced with modified values in the HIE to try and provide an approximation to a non-linear relationship between the T_{cloud} and derived intensity values. Two arrays containing T# values at the specific temperature breakpoint values in Table 5 are used to modify the regression equations for the Tcloud and base value parameters as shown in the following tables. The "original" column value is what the equation would have provided at the Tcloud temperature value, with the "modified" column displaying what the value was changed to.

	Eye Regression		Cloud Regression	
Temperature	Original	Modified	Original	Modified
+30°C	-1.00	1.00	2.00	2.00
+9°C	0.47	2.00	2.42	2.40
-30°C	3.20	3.25	3.20	3.25
-42°C	4.04	4.00	3.44	3.50
-54°C	4.88	4.75	3.68	3.75
-64°C	5.58	5.25	3.88	4.00
-70°C	6.00	5.75	4.00	4.10
-74°C	6.42	6.25	4.12	4.20
-80°C	6.70	6.75	4.20	4.30
-84°C	6.98	7.50	4.82	4.40
-100°C	8.10	8.00	4.60	4.70

Significant correlation improvements between HIE intensity estimates and aircraft reconnaissance intensity observations were realized with the implementation of the new regression equations over the previous intensity estimation techniques. The cloud scene correlations improved from 0.28 to 0.50, while the eye scenes improved from 0.62 to 0.70. The improvement in the cloud scene correlation is most significant since the CDO scene type still remains the most troublesome area within the HIE algorithm and one that continues to merit research attention.

3.4.2.5.3 Non-Regression Intensity Estimates

The remaining cloud scene types that do not utilize a regression-based intensity estimation scheme are the Curved Band and Shear scenes. Intensity estimates for these two scenes rely on techniques based on the original SDT. The Shear scene type measures the distance between the determined storm center position (usually the interpolated forecast position) and the leading edge of the main convective region defined as the minimum distance to the "Dark Gray" BD-curve Gray Scale temperature range (Table 5). The value of this distance is then related to intensity; the greater the distance the weaker the storm intensity. A distance of 140km or greater will correspond to a T# intensity of 1.5, with an increase in intensity of 0.5 for every 30km distance reduction (except for the following two exceptions; 1.) At 80km, the intensity is 2.25; and 2.).Between 0 and 35km, the T# intensity is at its maximum value of 3.5). Subsequent modifications have improved the identification of the edge of the main cloud region, leading to superior intensity results for this scene type.

The Curved Band scene type utilizes a methodology similar to the one used in the automated storm center determination scheme described in Section 3.3.2. This technique utilizes a 10-degree log spiral to measure the amount of convection and determines the amount of curvature, in terms of 15° segments, in the cloud region surrounding the storm center. The technique to compute the curvature measures the cloud top temperature at each point on the log spiral in 15° steps along the spiral. The measured temperature must be colder than the specific threshold temperature value being checked to be classified as convective cloud. The specific threshold temperature values correspond to the maximum temperature values within each gray scale temperature range listed in Table 5. The gray scale ranges between the Dark Gray and White enhancement ranges, inclusive are utilized in this process. Each point on the log spiral arc are checked at 15° increments, with a count of "consecutive cloud segments" retained to find the maximum number of consecutive segments in each gray scale range, if available. "Each point" on the spiral actually corresponds to the point and the surrounding eight points (a 3X3 grid centered on the log spiral point). If there are four or more points within the 3X3 grid with temperature values lower than the specific threshold temperature, it is counted as a valid point on the spiral. Consecutive cloud segments are counted on this arc to determine the amount of curvature (24 segments would correspond to 360-degrees of curvature).

The Light Gray temperature range is checked first, with warmer ranges checked if the number of consecutive segments does not exceed seven segments. If a gray range is not found in these three ranges, the scene type is assumed to be a shear scene. If the Light Gray region is found to possess more than 25 consecutive segments, the routine will then check the curvature of the Black and White temperature ranges. If each number of consecutive segments in the Black and White ranges are also greater than 25, the scene type will be assigned a CDO/Embedded Center scene type, otherwise the scene will be assigned a Curved Band scene type. The amount of curvature is then related to a T# intensity value based upon the original Dvorak Technique (with an additional adjustment provided for curved bands measure from the Black and White temperature ranges, described below). For less than 20% curvature the T# intensity will be equal to 1.5. Between 20% and 40% curvature, the T# range will increase from 1.5 to 2.5. Above 40% curvature the T# intensity will increase by 0.5, starting at 2.5 for 40% to 4.0 for 100% (and up to 4.5 for 120% curvature). T# values are interpolated to the nearest 0.1 increment between the curvature threshold values listed. If the curved band measurements are determined from the Black (White) temperature range, the T# value will be the minimum value between 4.0 (4.5) or the curvature value defined above plus 0.5 (1.0) T#.

The storm center utilized for the Curved Band scene is primarily the interpolated official forecast since the storm strength is typically too low for the auto-centering techniques to be used. The 10-degree log spiral is rotated in 10-degree increments around the storm center to find the maximum cloud curvature amount at the selected center position. A search for the storm center position that yields the highest value is also conducted and output to the user for reference if it does not correspond to the original, interpolated forecast position.

3.4.2.6 Intensity Estimate Conversion: T#/CI# to MSLP/Maximum Wind

The HIE intensity estimates are derived in terms of T# and CI#, which are a holdover from the SDT, where higher values correspond to stronger, more intense tropical cyclones. The T# is short for "Tropical Number", and is defined in the SDT in relation to storm strength categories of mean sea level pressure (MSLP) and maximum sustained surface wind speeds (Dvorak 1984). The T# represents the "current" intensity of the storm. There are three T# values derived by the HIE; the Raw T#, the Adjusted Raw T#, and the Final T# values. These are explained in the following section (Section 3.4.2.6).

The Current Intensity (CI#) is derived from the HIE Final T# value after a final set of time-based intensity constraint rules/adjustments are applied after the storm has stopped (temporarily?) strengthening, and represents the final intensity estimate output by the HIE. The three T# and CI# values can differ depending on the current intensity status of the storm and how it has changed over a given period of time (the constraint rules can vary over different time periods), but each value provides information about the current storm status and evolution.

The empirically-derived relationship between the CI# and storm intensity categories is given in Table 6 for the Atlantic TC basin (after Dvorak, 1984). The Northwest Pacific basin relationships (Shewchuck and Weir, 1980) differ slightly and are utilized in all basins outside of the North Atlantic. Thus, proper identification of the correct relationship to use is essential for conversion of the CI# to intensity. The HIE can automatically determine the conversion method "basin" based upon the longitude value of the selected storm center location. All storms within the Atlantic basin (east of the US/Mexican/South American east coast (including the Carribean Sea) and west of the Prime Meridian (0° E/W) and African coast will use the Atlantic conversions, while all other basins will use the West Pacific conversions (including the East/Central Pacific and all basins in the Eastern Hemisphere).

The T#/CI# values in Table 6 are provided in steps of 0.1 in the HIE algorithm outputs, with the corresponding MSLP/Maximum wind values linearly interpolated between the T#/CI# node values in Table 6. All T#/MSLP/Wind values are stored within the HIE using a series of arrays. When converting between units the derived value is not derived directly using a linear equation, instead the array values are used to determine the current intensity at T#/CI# 0.1 resolutions. For example, if the conversion is done from Wind/MSLP to T#, a linear interpolation could be performed, but the T# value is truncated at the tenths digit, and it is never rounded up (4.58 would NOT be 4.6… it is 4.5). The conversion methodology in the HIE is to search the Wind/MSLP array and find the largest/smallest value (index) that is less/greater than the input Wind/MSLP value. This array index is then used to determine the T# value. For example, if the input wind speed is 75.0 knots, the Wind Array would be scanned. The T# of 4.4 has a corresponding (interpolated) Wind Speed value of 74.6, while the T# of 4.5 has a corresponding Wind Speed value of 77.0 knots. The T# value corresponding to a speed of 75.0 would be 4.4.

It should be noted that for T#/CI# values greater than 8.0 or less than 2.0 in Table 6, the MSLP estimates were not provided by the original Dvorak study, and are thus extrapolated in the HIE by using MSLP gradients from preceding/following categories in the conversion table. These values include those found between 1.0 (nclusive) and 2.0 (exclusive) and between 8.0 (exclusive) and 9.0 (inclusive).

Classification (approx)	CI# Value	Wind Speed(kts)	Atlantic MSLP(hPa)	Pacific MSLP(hPa)
	1.0	*25.0*	*1014.0*	*1005.0*
	1.5	*25.0*	*1012.0*	*1003.0*
	2.0	30.0	1009.0	1000.0
Tropical Storm	2.5	35.0	1005.0	997.0
	3.0	45.0	1000.0	991.0
	3.5	55.0	994.0	984.0
Hurricane – Cat 1	4.0	65.0	987.0	976.0
	4.5	77.0	979.0	966.0
Hurricane – Cat. 2	4.7	82.2	975.4	961.2
	5.0	90.0	970.0	954.0
Hurricane – Cat 3	5.2	94.8	966.0	948.8
	5.5	102.0	960.0	941.0
Hurricane – Cat 4	6.0	115.0	948.0	927.0
	6.5	127.0	935.0	914.0
Hurricane – Cat 5	6.8	134.8	926.6	904.4
	7.0	140.0	921.0	898.0
	7.5	155.0	906.0	879.0
	8.0	170.0	890.0	858.0
	8.5	*185.0*	*873.0*	*835.0*
	9.0	*200.0*	*855.0*	*810.0*

Table 6: Relationships between HIE-derived CI# values and TC maximum sustained surface wind speed and mean sea level pressure (MSLP) values for Atlantic and West Pacific basin storms.

3.4.2.7 Intensity Constraint Limits (Dvorak EIR Rule 8)

The HIE limits the magnitude of intensity change between estimates based on a modification to the SDT EIR Rule 8 which constrains the rate of increase or decrease of the Raw T# intensity estimate (value initially determined by the scene type and/or regression analyses, and before any rules or adjustments are applied) based upon specified rates of change over designated time periods. The rate of change checks are performed against the Final T# (after rule adjustments and time averaging described later) values stored in the history file for times prior to the current analysis time. The rate of change allowance is dependent upon the current intensity of the storm, as defined in the Dvorak EIR technique (Dvorak 1984). The Final T# value of the record immediately prior to the current analysis time is used as the current intensity estimate. When the intensity of the storm is less than T4.0, the current Raw T# rate of change cannot exceed ±0.5 T# over 6 hours. If greater or equal to T4.0, the rate of change of the Raw T# value cannot exceed the following rates: 1.0 over 6 hours, 1.7 over 12 hours, 2.2 over 18 hours, and/or 2.7 over 24 hours.

The HIE also employs a modified version of this rule based upon the current scene type analyzed: The allowable rate of change is reduced by up to 0.5 during most non-Eye scene types (Curved Band, CDO, Embedded Center), so that the amount of change is limited to 0.7/1.2/1.7/2.2 T# over 6/12/18/24 hours. For Eye scenes the amount of allowed change is increased by up to 1.0 to 1.7/2.7/3.2/3.7 T# over 6/12/18/24 hours. The latter modification allows the Final T#/CI# to increase at a quick enough and realistic rate during "rapid intensification" events. For Shear scenes, the rate of change rule is unmodified from the original SDT EIR Rule 8.

If one of the criteria is exceeded, the Adjusted Raw T# (value after all rules/adjustments, but before time-averaging) will be assigned the maximum/minimum value allowed by the above criteria, and will be used in the determination of the corresponding Final T# and CI# values. Note: The unadjusted Raw T# value is always stored in the history file for reference to the raw intensity estimates before any adjustments.

As a final "gross error check" for the Raw T# values derived by the HIE, a supplementary Rule 8 rule has been added to the constraint limits listed above. This additional rule will limit the growth of the Raw T# value by 0.5 over 1 hour. Implementation of this rule was found to be necessary due to the inherent variability of the HIE during "scene type transitions", especially between eye and non-eye cases. Due to the completely objective nature of the scene determination scheme, and the discrete threshold values that are used to define the different scene types, changes between scene types can result in unrealistic jumps in Raw T# values from one intensity estimate to another. The addition of this rule produces estimates that are reasonable when these types of scene changes occur, but will not allow the Raw T# values to change at a rate greater than observed in nature.

Implementation of this modified rule has allowed for the Final T# calculation to be changed from a time-weighted 6-hour average to a 3-hour average (discussed in the next section). By utilizing the 3-hour scheme, the admissible rate of change of the Final T# value is greater than with the 6-hour scheme, making for more allowable rapid increases in intensity over short time periods that can be associated with storms undergoing rapid intensity changes.

3.4.2.8 Time Averaging Scheme

Time-averaging of HIE intensity outputs is employed to smooth the "noise" inherent in individual estimates, which as mentioned above can result from objectively changing scene types and crossing value thresholds in consecutive analysis times. The time-averaging scheme uses the current adjusted Raw T# and all available adjusted Raw T# values obtained within the last 3 hours. The time- average value is a straight non-weighted mean of these estimates and is considered for estimates only derived over open water (not over land). The 3-hr time window for the averaging is based on empirical analysis noted above, and user preference input. The HIE Final T# at any current analysis time reflects this time-averaged value. If no records are found within the 3 hour averaging period, the current adjusted Raw T# value is utilized as the "averaged value".

3.4.2.9 Weakening Flag (Dvorak EIR Rule 9)

The HIE Weakening Flag is modeled after the STD EIR Rule 9 (Dvorak, 1984), but also utilizes a modification to the original STD EIR Rule 9 that is applied at some operational TC forecasting centers. Weakening is defined as a negative difference between the current intensity and previous final T# values. The “original” STD ERI Rule 9 rule is used in the determination of the CI# after a storm has reached its maximum intensity and is weakening during the dissipation stage of the TC lifecycle (e.g. extra-tropical transition, moving over cold waters, interacting with severe atmospheric shear, etc). The EIR Rule 9 will hold the CI# at values 0.5 to 1.0 higher in value than the current Final T# (Raw T# after adjustments and time-averaging). Subjective application of this rule (e.g. how and when to apply it) varies between forecasters and is the focal point of much debate. The value of 1.0 is utilized in the HIE since it provides the most statistically accurate estimates of storm intensity when compared with actual reconnaissance observations.

The “modified” Weakening Flag rule logic affects the calculation of the CI# during the formation/strengthening stage of the storm prior to maximum TC intensity (where the ”original” Rule 9 rule will be invoked) as follows: “Always hold the CI# to the highest Final T# over the last 6 hours, but never greater than 1.0, in all cases”. This rule will hold the HIE CI# equal to the highest Final T# obtained during the previous 6 hours. For example, if the Final T# value had previously obtained a value of 5.2, but now is decreasing, the CI# value will be held at 5.2 for up to 6 hours or until the Final T# falls below 4.2. This rule is dis-invoked if the TC re-strengthens to the held CI# and greater intensity values.

If a storm is over land for greater than 12 continuous hours, the application of the Weakening Flag will be discontinued if/when the storm reemerges over open ocean, with the CI# value set to the Final T# value. If the storm is over land for greater than 24 continuous hours, the storm intensity is reinitialized to the input “Initial Classification” value input into the HIE algorithm (derived from the wind speed value from the TCFC forecast).

3.4.2.10 Rapid Weakening Flag

It is commonly observed that TCs can dissipate at a very rapid rate as they move over colder waters. However, the Dvorak Technique Rule 8 and Rule 9 constrain the rate of dissipation and it is believed that these rules do not allow for the rapid dissipation rates commonly observed in the East Pacific region (and periodically noted in other oceanic regions).

In order to alleviate this limitation, the Dvorak EIR Weakening Rule 9 has been modified to allow the CI# intensity estimate to reflect this commonly observed phenomenon. The HIE Rapid Weakening Flag relaxes the application of the original Rule 9 intensity adjustment to the Final T# when determining the CI#. Normally, up to 1.0 T# is added to the current Final T# value to come up with the CI#. In cases where rapid weakening is determined, the additive value is modified to 0.5. To be identified as a possible rapid weakening event, the Adjusted Raw T# values over the previous six hours are evaluated. If the slope of these values is less than -0.5/6 hours the weakening flag is tripped and subsequently observed for an additional six hours. If over a second six hour period the slope does not increase over the threshold, the rapid weakening flag is turned on and the CI# additive constant is adjusted from 1.0 to 0.5. The rapid weakening flag is held on until the slope of the Adjusted T# values over six hours becomes greater than -0.37/6 hours. If this condition is exceeded for a continuous period of six hours, the rapid weakening flag is then turned off and the additive value returns to 1.0.

3.4.2.11 MSLP Latitude Bias Adjustment

If the Knaff/Zehr pressure/wind relationship is not utilized by the HIE (Section 3.3.2.5), the MSLP value output by the HIE will include an adjustment based on the latitude of the storm, designed to eliminate a bias noted by Kossin and Velden (2003) in the HIE-derived MSLP values and underlying Dvorak Technique. The equation used for this adjustment is:

$$CI\#_{adj} = CI\#_{orig} + (7.325 - (0.302)*ABS(latitude)))$$

This equation requires the $CI\#_{orig}$ value to be entered in units of pressure (hPa). Since the HIE produces estimates in terms of T# values, the $CI\#_{orig}$ value must be converted to millibars before the adjustment equation is applied. The pressure adjustment value is stored in a history file.

The MSLP bias adjustment is only applied to scene types that utilize eye and cloud temperature values to determine the strength of the storm since the bias is related to corresponding temperatures associated with a higher/lower tropopause height. In order for the adjustment to be turned "on", six hours of continuous "EIR-type" scenes must be present. These types of scenes are CDO, Embedded Center, and Eye scenes, while non-EIR scenes include Shear, Curved Band, and Irregular CDO. Once the scene criterion is met, the bias adjustment is blended into use over the next six hours using a time-weighted multiplication factor, ranging from 0.0 at the initial time to 1.0 at the six hour time period, with the determined latitude bias adjustment value.

For more specific information about the latitude bias adjustment, please refer to the following online document (Kossin and Velden, 2003) at:

http://www.ssec.wisc.edu/~kossin/articles/kossin_velden_MWR.pdf

3.4.2.12 Radius of Maximum Wind Estimate

The HIE invokes a subroutine after each intensity estimate to approximate the radius of maximum wind (RMW). This subroutine calculates the RMW using a regression equation derived from historical comparisons between IR imagery cloud top temperature fields and collocated in-situ reconnaissance aircraft radius of maximum wind measurements. If the derived HIE cloud region temperature (described in Section 3.4.2.3) is colder than –50°C, the distance from the storm center to the -45°C cloud top temperature isotherm is measured in four directions and averaged to produce a final distance measurement. The RMW location (in kilometers from the storm center position) is then determined from this distance using the equation:

$$\text{RMW} = 2.8068 + (0.8361*\text{dist})$$

If the HIE derived cloud top temperature value is equal to or warmer than -50°C, the distance to the isotherm defined by the equation (ERT + 2.0*CRT)/3.0, where ERT is the eye (center) temperature and CRT is the cloud region temperature, is used in the above equation to determine the RMW value.

3.4.2.13 Passive Microwave (PMW) Eye Score

In order to try and alleviate a long standing issue with obtaining accurate intensity estimates with the HIE during Central Dense Overcast (CDO) scene types, an external routine, using passive microwave (PMW) imagery, is used to derive an "eye score" value that the HIE can use to obtain a more accurate intensity estimate than can be obtained with the IRW imagery alone. The process of how the PMW eye score is derived will be detailed in a yet to be written document.

Once the eye score is derived, the score and the PMW scan date and time information is passed into the HIE algorithm. Typically the eye score information is delayed from the current IR analysis time by anywhere from one hour to longer delay periods. This does not pose a problem for the HIE.

Utilization of the PMW eye score value will only be allowed prior to three consecutive, automated "eye" scene type determinations by the HIE algorithm. Once three consecutive eye scene types have been found using the IR imagery alone, the PMW "eye score" information will be ignored no matter what its value may be.

Before the three eye scenes are found, the PMW eye score value will be used to determine the intensity of the storm at the time of the PMW eye score value (it will actually be assigned to the time record in the existing storm history file that exists immediately after the date/time of the PMW eye score record). There are two threshold eye score values that are utilized in the PMW eye score routine, each corresponding to a different T# intensity value to be assigned in the HIE algorithm for the intensity. If the eye score value is equal to or greater than 20.0, but less than 60.0, the HIE will reassign the Raw T# intensity score of the existing record in the HIE history file to be 4.3. If the eye score value is greater than or equal to 60.0, the Raw T# value for this record will be assigned to a 5.0. If no records exist in the HIE history file after the PMW eye score record time, the intensity estimate for the current IR image being analyzed will be modified to one of these two values.

In order to alleviate an artificial "jump" in the HIE history file records once the PMW eye score value exceeds one of the threshold values, all of the CI# values, as well as the adjusted Raw T# and Final T# values, in each record up to 12 hours prior to the PMW eye score record time will be adjusted linearly to a value between the "current" PMW eye score intensity value and the CI# value of the 12-hour previous record. The history file records will be physically changed within the history file, but the "historical" values (obtained prior to the application of the PMW eye score adjustment) will be kept in the history file comment section for each record.

For example, if the PMW eye score value of 26.2 occurred at 20:56UTC, the following record in the history file (most likely at 21:15UTC) will have its Adjusted Raw T#, Final T#, and CI# values all changed to 4.3. Looking back 12 hours from the 21:15UTC record shows the 9:15UTC record with an intensity value (CI#) of 3.4. Thus, all records between 9:15 and 21:15 will have their Adjusted Raw T#, Final T#, and CI# values changed to the corresponding linear interpretation value between 3.4 and 4.3 (depending on the time between the two end times). The original values will be stored for each of these times in their corresponding comment section in the history file.

The same adjustment 12 hour previous record adjustment will be applied if the PMW eye score values change from the first threshold range (between 20.0 and 59.9) to the second (60.0 and above). For example, if the MW eye score exceeds 60.0 at 14:45UTC, all records back to 2:45UTC will be linearly adjusted between the 5.0 value at 14:45 and the CI# value at 2:45UTC.

The PMW eye score adjustment will continue to be applied as long as two conditions are not met. The first, as mentioned previously, occurs when three consecutive, automatically determined scene types are determined from the IR imagery to be "eye" scene types. The second occurs when the PMW eye score value drops below the 20.0 threshold value for more than 12 hours. Once either of these situations occurs, the PMW eye score adjustment will be terminated. If the first situation occurs, the PMW eye score adjustment can never be applied during the remainder of the storm's lifecycle. If the second situation occurs, the PMW adjustment can once again be applied if a PMW eye score value once again exceeds a threshold value.

During the application of the PMW eye score adjustment, the Rule 8 flag value in each history file record will be modified to store a descriptive value for each situation during the PMW eye score adjustment period. These values will be displayed in the History file Listing described in Section 3.3.2.1. The value will be "MW ON" when the eye score values are 20.0 and above, "MW Adj." for the 12 hour period prior to the first eye score values exceeding 20.0, and "MW Off" for either of the two conditions where the eye score adjustment is turned off (the records 12 hours previous to the "MW Off" will be assigned “MW Hold” in the second condition described in the above paragraph). The records modified during the initial 12-hour adjustment periods (either with the first eye score value above 20.0 or with the transition between the 20.0 and 60.0 values) will all have their comment section appended with the information “MWinit=A.A/B.B/C.C”, where A.A, B.B, and C.C are the initial Adjusted Raw T#, Final T#, and CI# values, respectively (prior to the MW adjustment).

3.4.3 Algorithm Output

Filename	Format	Contents
User determined	netCDF	History file (HIE analysis parameters)
User determined	netCDF	HIE history file listing
User determined	netCDF	HIE output error and diagnostic codes
User determined	netCDF	HIE Metadata - Storm Name and Number w/ basin ID (e.g. 03L) - Storm Position (latitude and longitude) - Current Intensity - Difference between current and previous intensity - Number of Measurements during storm history

Table 7: Output data file from HIE algorithm

The primary product output for the HIE algorithm is the history file, which contains all of the important derived analysis information for each individual image analysis. These parameters are listed in Section 3.3.2.1, but they will be described in the tables in Section 3.4.3.1 in terms of product, quality, and diagnostic parameters.

HIE runtime error and diagnostic codes will be presented in Section 3.4.3.2. These codes will be presented in a bit/byte format for each of the different codes provided during run-time execution of the HIE algorithm. Error codes will provide information when the HIE is not able to execute to completion successfully (output an intensity estimate), while the diagnostic codes provide important information to the user during the intensity estimation process which do not result in termination of the algorithm.

HIE metadata, described in Section 3.4.3.3 are the required HIE output parameters to meet algorithm specifications which will be output to the framework netCDF file.

3.4.3.1 History File Output

The following parameters described in Tables 8, 9, and 10 will describe the Product, Quality Flag, and Diagnostic output parameters contained within the HIE history file output for each analysis during each storm lifecycle. The official F&PS HIE required output is wind speed. This value is not directly output in the HIE history file, but instead it is derived from the CI# value as described in Section 3.4.2.5 and listed in Table 6.

3.4.3.1.1 Product Output

Category	Description	Value
CI#	Current Intensity number (wind speed value derived from CI# using relationship in Table 6)	1.0 – 9.0

Table 8: HIE History File product output value

3.4.3.1.2 Quality Output

Category	Description	Value
Land/Ocean flag	Storm center position (latitude and longitude) over land or ocean	1=Over Land 2=Over Ocean

Table 9: HIE History File quality output value

3.4.3.1.3 Diagnostic Output

Category	Description	Value
Date	Image analysis date	YYMMMDD format (YYYY=year, MM=3 letter month ID, DD=day)
Time	Image analysis time (in UTC)	hhmmss format (hh=hour, mm=minutes, ss=seconds)
Julian date and partial date	Julian date and percentage of day at image analysis time	jjj.pppp (jjj=julian date, pppp=partial date %)
Raw T#	Raw T# value	1.0 – 9.0
Raw T# adjusted	Adjusted Raw T# value	1.0 – 9.0
Final T#	Final T# value (3 hour time average)	1.0 – 9.0
Eye Temperature	Warmest temperature value in eye region (0-24km from storm center)	In degrees Celsius
Cloud Temperature	Average cloud top temperature in 80km wide annulus centered at storm center (minimum radius=24km, maximum radius=136km)	In degrees Celsius
Cloud Temperature (old value)	Average cloud top temperature between 24 - 136km from storm center	In degrees Celsius
“Coldest-Warmest” Cloud Temperature	Warmest temperature on coldest 4km ring between 24 – 136 km from storm center	In degrees Celsius

Latitude	Storm center latitude	North = positive
		South = negative

Longitude	Storm center longitude	West = positive
		East = negative

Eye/CDO size or	Value containing either the radius of	In kilometers

shear distance	the storm eye feature, the size of the cloud CDO region, or the distance from the storm center to the cloud edge (dependent upon eye and cloud scene type values)	
Eye StdDev	Standard deviation of the eye region temperature values	In degrees Celsius
Cloud Symmetry	Cloud region symmetry value (average difference between opposing cloud region slices in annulus)	In degrees Celsius
Satellite ID	Satellite ID value	n/a
Eye Scene Type	Eye region scene type	0 = Clear Eye 1 = Pinhole Eye 2 = Large Eye 3 = No Eye
Cloud Scene Type	Cloud region scene tyoe	0 = CDO 1 = Embedded Center 2 = Irregular CDO 3 = Curved Band 4 = Shear
Rule 8 Flag	Dvorak EIR Rule 8 constraint limits	Ten's designations 0's=Shear Scene 10's=Eye Scenes 20's=Curved Band/CDO/EmbC/ IrrCDO Scenes 30's MW Adjustments : 30=12-hr Interpolation 31=Initial MW adjustment 32=MW adjustment On 33=Holding MW adjustment 34=MW adjustment Off One's designations 0=No Constraint 1=T#4.0 0.5/6hr 2=6hr rule 3=12hr rule 4=18hr rule 5=24hr rule 8=0.2/hr (1st six hours only), 9=0.5/hr
Rule 9 Flag	Dvorak EIR Rule 9 weakening flag	0 = Off 1 = On 2 = On (Initialized at >-6.0)
Latitude Bias	Latitude Bias application flag	0 = Off 1 = Merging (12 hours) 2 = On
Rapid Dissipation	Rapid Dissipation flag (only applied	0 = Off

	in East Pacific ocean basin < 140W)	1 = On
Eye FFT	Eye region FFT harmonic value	0 – 13
Cloud FFT	Cloud region FFT harmonic value	0 – 13
Curved Band Gray Scale	BD enhancement Gray Scale value	See Table 5
Curved Band Curvature	Amount of curvature in Curved Band Gray Scale analysis region	X = 1 – 25 (curvature % = X-1/24)
Coldest Warmest Distance	Distance to "coldest-warmest" ring from storm center	In kilometers
Fix Method	Automated storm centering method	0 = manual 1 = forecast interpolation 4 = spiral analysis 5 = ring/spiral combination 6 = extrapolation
Latitude Bias Adjustment	Amount of latitude bias adjustment to MSLP value (from MSLP to T# relationship)	In millimars
RMW	Derived radius of maximum wind from satellite imagery	In kilometers
PMW value	External PMW "eye score" value	n/a
Gale Radius	34 knot wind speed gale radius value (for use in Knaff/Zehr MSLP/Wind speed relationship)	In kilometers
Environmental Pressure	Environmental storm pressure value (for use in Knaff/Zehr MSLP/Wind Speed relationship)	In millibars

Table 10: HIE History File diagnostic output values

3.4.3.2 Output Error and Diagnostic Codes

The HIE performs validity checks for the various input products and internally determined values during the execution of the HIE. Various error and diagnostic messages are produced during the execution of the HIE and output at the end of the processing cycle. These messages can be used by the user to determine why a particular HIE run failed or how the HIE obtained a certain intensity estimate value. These error and diagnostic values are shown in Tables 11 and 12. The error messages are assigned the error bit values as global variables when the code exits and the error message is printed to the screen. These values are then turned into a single output value by byte shifting them and then combining the two values with the following equation (values are as listed in the error code table enclosed below; error code 1 is the first column, error code 2 is the second column):

Output error code value = Error Value 1*256 + Error Value 2
Where: Error Value 1 = 1<<(error code 1) and Error Value 2 = 1<<(error code 2)

GOES-R HIE Error Codes

Byte	*Bit*	*HIE Code*	*Error Description*
0	**0**	-1	Error reading from existing history file Error reading data record structure in history file (sscanf error)
0	**1**	-2	Error creating history file History file could not be created (system error)
0	**2**	-3	Error writing to history file Error writing data record structure to history file (fprintf error)
0	**5**	-152	Empty
0	**6**	-153	Empty
0	**7**		Empty
1	**6**	-17	Data read off edge of image Storm center is too close to edge of image; image reading is trying to read off top/bottom/left/right side of image). Also this error will result when there are too many "bad lines" in image (more than 10 lines; a bad line has more than 10 bad pixels in a line). Typically "bad lines" happen if image is blank (has not loaded correctly or filled completely) but navigation is still valid for the image)
1	**7**		Empty
2	**2**	-23	Bad navigation Error converting from line/element to latitude/longitude when trying to read image file
2	**3**	-31	Error accessing topography file Can't find topography file (fopen error)
2	**4**	-32	Error reading topography file Can't read topography file (fread/fseek error)
2	**5**	-41	Fourier Transform Analysis failure Error computing FFT analysis of image
2	**6**	-51	Error with eye and/or cloud region temperature value(s) Eye or cloud region tempeatures <-100C or >+40C
3	**0**	-43	Error accessing forecast file; extrapolation failed Could not read forecast file (fopen error)
3	**1**	-44	Invalid forecast file; extrapolation failed Forecast file was of wrong type or file type is not valid
3	**2**	-45	Error reading forecast file; extrapolation failed Error reading forecast file (fread error)
3	**3**	-46	Forecast interpolation failure; extrapolation failed Could not interpret to image time using data in forecast file
3	**5**	-72	Empty
3	**6**	-81	Error determining ocean basin Could not set ocean basin because latitude/longitude values were out of range (not within +90/-90 and -180/+180)
3	**7**		Empty
4	**7**		Empty

Table 11: HIE algorithm runtime error codes.

GOES-R HIE Diagnostic Codes

Byte	***Bit***	***HIE Code***	***Error Description***
0	**0**	-1	Error reading from existing history file Error reading data record structure in history file (sscanf error)
0	**1**	-2	Error creating history file History file could not be created (system error)
0	**3**	-4	Error writing comment to history file (date error) Could not find date specified when trying to add comment to existing record (using DATE keyword)
0	**5**	-152	Empty
0	**6**	-153	Empty
0	**7**		Empty
1	**0**	-11	Error reading image file Error opening image file using McIDAS opnara function (or from mcidas_ routines that are being replaced by AIT in HIE)
1	**1**	-12	Error accessing image file Error reading image file using McIDAS readd function (or from mcidas_ routines that are being replaced by AIT in HIE)
1	**6**	-17	Data read off edge of image Storm center is too close to edge of image; image reading is trying to read off top/bottom/left/right side of image). Also this error will result when there are too many "bad lines" in image (more than 10 lines; a bad line has more than 10 bad pixels in a line). Typically "bad lines" happen if image is blank (has not loaded correctly or filled completely) but navigation is still valid for the image)
1	**7**		Empty
2	**2**	-23	Bad navigation Error converting from line/element to latitude/longitude when trying to read image file
2	**3**	-31	Error accessing topography file Can't find topography file (fopen error)
2	**4**	-32	Error reading topography file Can't read topography file (fread/fseek error)
2	**5**	-41	Fourier Transform Analysis failure Error computing FFT analysis of image
2	**6**	-51	Error with eye and/or cloud region temperature value(s) Eye or cloud region tempeatures <-100C or >+40C
3	**0**	-43	Error accessing forecast file; extrapolation failed Could not read forecast file (fopen error)
3	**1**	-44	Invalid forecast file; extrapolation failed Forecast file was of wrong type or file type is not valid

3	**2**	-45	Error reading forecast file; extrapolation failed Error reading forecast file (fread error)
3	**3**	-46	Forecast interpolation failure; extrapolation failed Could not interpret to image time using data in forecast file
3	**5**	-72	Empty
3	**6**	-81	Error determining ocean basin Could not set ocean basin because latitude/longitude values were out of range (not within +90/-90 and -180/+180)
3	**7**		Empty
4	**7**		Empty

Table 12: HIE algorithm runtime diagnostic codes.

3.4.3.3 HIE Metadata

The HIE will be required to output a series of metadata values. These values will include the following parameters:

Storm Name and Number (including official basin ID value) **Storm Number:** Number of storm in ocean basin during current storm season for basin (e.g. first storm of season is given the number 01, the second is given the number 02, etc.) **Basin ID values:** L=Atlantic, E=East Pacific (east of 140°W to US/Mexican coast), C-Central Pacific (between 140°W and dateline) **Example:** the third storm in the Atlantic basin would be 03L; the twelfth storm in the East Pacific basin would be 12E)
Storm location (latitude and longitude)
Current storm intensity (F&PS requirement is wind speed in m/s)
Difference between current and previous storm intensity (wind speed in m/s)
Number of intensity measurements for entire storm history

Table 13: HIE algorithm metadata values

These values will be provided to the user along with any other output values, such as error codes, and tailored products such as storm listings and current intensity bulletins.

4 TEST DATA SETS AND OUTPUTS

4.1 AVHRR and MODIS data validation study

4.1.1 Simulated Input Data Sets

The initial test of the HIE algorithm (Version 5) on GOES-R ABI-like data, a GOES-R proxy dataset consisting of 1km spatial resolution AVHRR and MODIS images was collected and archived by the Cooperative Institute for Research in the Atmosphere (CIRA) at Colorado State University. This dataset consists of the following imagery sets:

- Polar satellite (AVHRR or MODIS) full resolution imagery
- Time-matched (as close as possible) full resolution GOES IR
- AVHRR or MODIS IR remapped to 1km (*P01*)
- Reduced resolution polar IR 2km (resembling the GOES-R footprint) (*P02*)
- Reduced resolution polar IR 4km (resembling the current GOES footprint) (*P04*)
- GOES IR remapped 4km (*G04*)

A full description of the datasets can be obtained from the *GOES-R AWG Winds Group Tropical Cyclone Proxy Data and Intensity Estimation Algorithm Documentation and Preliminary Results* report, which can be found at the CIRA AWG project website at the following address: http://rammb.cira.colostate.edu/projects/awg.

For HIE testing, only the full resolution Polar imagery and the GOES IR remapped 4km data is used. Eleven TCs were selected for testing consisting of a total of 357 individual scenes which occurred between 2002 and 2006 (Table 14). Each of the polar images uses the full resolution IR data from MODIS or AVHRR and remaps it to a TC-centered Mercator projection image with 1 km resolution.

Storm Name & Year	Scenes	Size (MB)
Lili 2002	21	150
Isabel 2003	66	469
Emily 2005	41	286
Katrina 2005	57	377
Rita 2005	43	303
Stan 2005	13	93
Wilma 2005	61	435
Alpha 2005	8	58
Beta 2005	9	66
Hilary 2005 (EP)	12	87
Ernesto 2006	26	187
Total	**357**	**2511**

Table 14: TC case study sample information used for testing the HIE on GOES-R proxy data.

HIE intensity estimates are processed using the collocated GOES-IR 4km spatial resolution data from the current GOES satellites and the GOES-R proxy data from the polar data in order to assess potential impact on the HIE intensity estimates from the improved ABI spatial resolution. HIE intensity estimates are only derived using the collocated images for each dataset (closest match temporally between the current GOES and polar/GOES-R proxy data sets).

Figure 11 illustrates the improvement realized with the simulated GOES-R ABI imagery over the current GOES-12 imagery for Hurricane Wilma on 19 October 2005. Wilma possessed an extremely small eye at this time. HIE intensity estimates using the GOES-12 infrared window (IRW) drastically underestimated the storm intensity at this time due to the inability to resolve the eye (and thus led to an incorrect HIE scene type determination). Using the simulated GOES-R ABI imagery, the HIE was able to identify the scene type correctly as an eye feature, resulting in a much more accurate intensity estimate for this individual image.

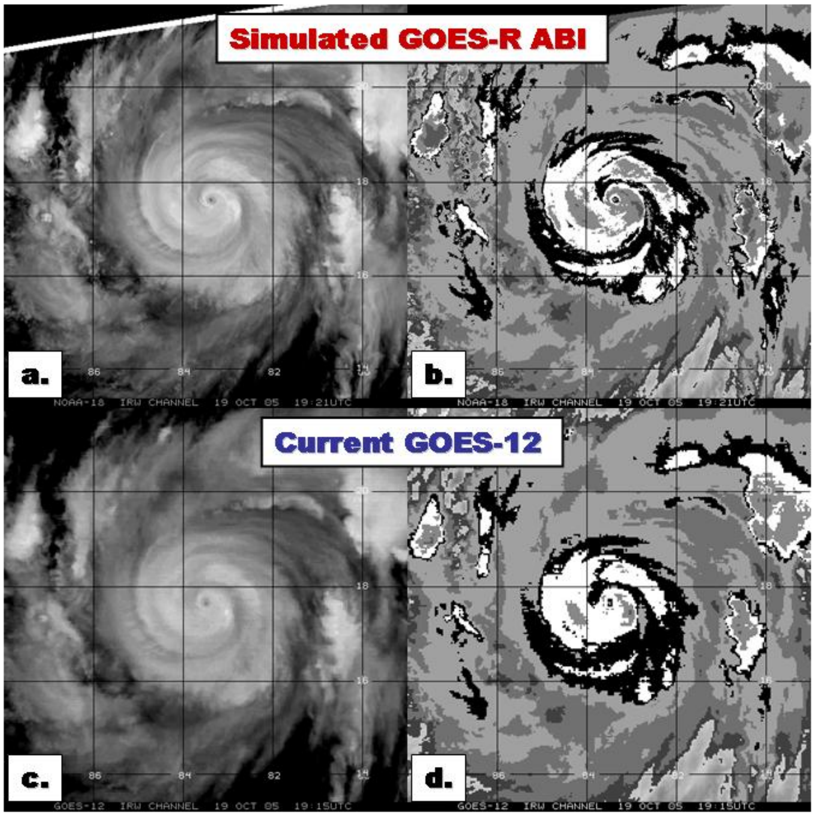
Simulated GOES-R ABI
Current GOES-12
a.
b.
c.
d.
20
18
16
86
84
82
NOAA-18 IRW CHANNEL 19 OCT 05 19:21UTC
GOES-12 IRW CHANNEL 19 OCT 05 19:15UTC

Figure 11: Simulated GOES-R ABI infrared window (IRW) imagery with a.) black/white contrast stretch and b.) "BD-curve" enhancement, and current GOES-12 IRW imagery with c.) black/white contrast stretch and d.) "BD-curve" enhancement, for Hurricane Wilma on 19 October 2005 (19:21 and 19:15UTC,

respectively). Contrast stretch/enhancement show improved capability to capture small eye feature with the ABI.

4.1.2 Output from Simulated Input Data Sets

As discussed in the previous section, the ability to discern smaller TC features, such as extremely small “pinhole” eyes using higher spatial resolution imagery, can lead to improved intensity estimates from the HIE algorithm. Examination of the HIE intensity estimates for Hurricane Wilma in 2005 over the entire storm lifetime illustrates the impact on numerous HIE intensity estimates, especially during the extremely small eye time period. Figure 12 shows the HIE Raw T# value differences derived from the current GOES and GOES-ABI, versus verifying aircraft reconnaissance. The intensity differences are most notable during the period from about 19 October/18:00UTC to about 20 October/12:00UTC. During this period the HIE had significant problems resolving/retaining an eye feature scene type when utilizing current GOES-12 IR imagery. When utilizing the GOES-R ABI data, the HIE was able to resolve and maintain an eye scene consistently during the same time period, resulting in intensity estimates more closely matching those obtained by aircraft reconnaissance.

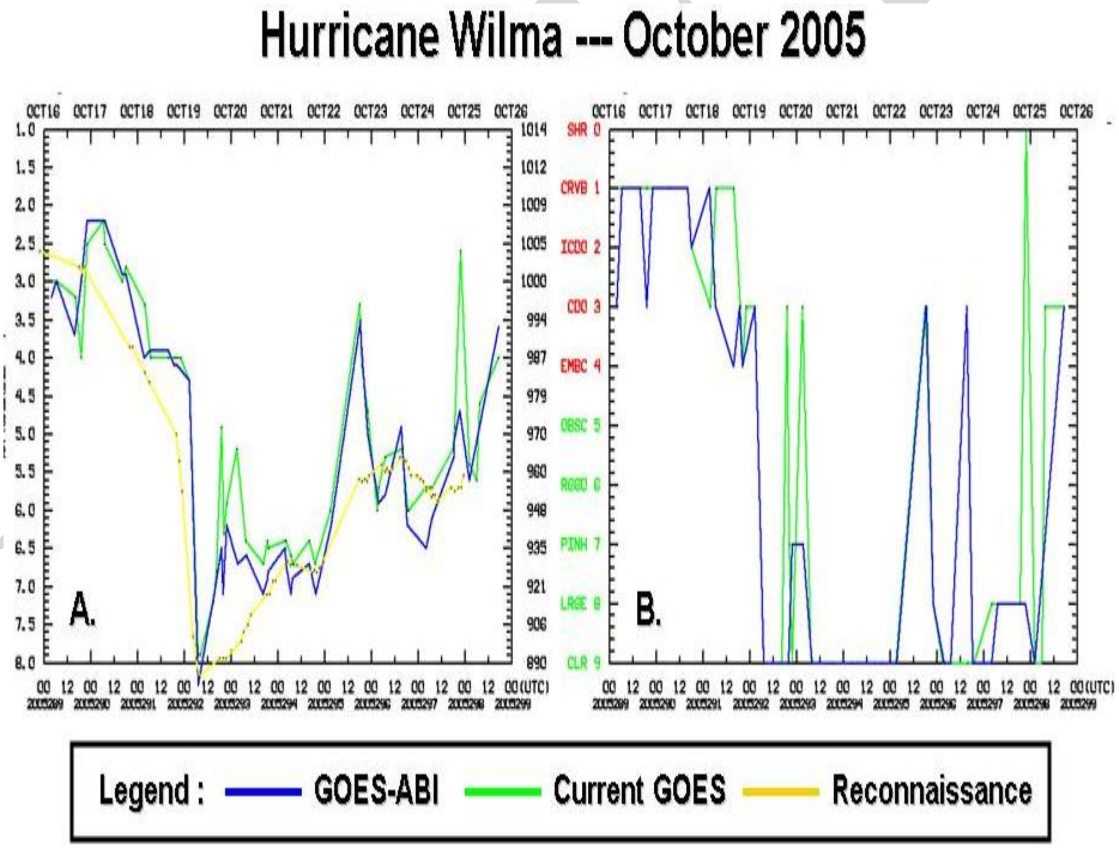

Figure 12: HIE Raw T# Intensity Estimates and B.) Scene type selection for Hurricane Wilma (2005). GOES-ABI (blue), current GOES (green), and aircraft reconnaissance (yellow) values are displayed.

4.1.3 Precision and Accuracy Estimates

Homogeneous HIE intensity estimates from the current and simulated GOES data sets are compared using a total sample of 39 estimates within one hour of a verifying reconnaissance wind speed measurement over 11 Atlantic basin storms between 2002 and 2006. Table 15 shows the statistical comparisons of the HIE intensity estimates for both the current and simulated GOES data sets using Version 5 of the HIE.

	GOES-ABI		GOES-Current	
Intensity	Accuracy (m/s)	Precision (m/s)	Accuracy (m/s)	Precision (m/s)
CI#	-3.56	7.57	-3.83	7.14
OpCen	-2.09	6.23	-2.09	6.23

Table 15: HIE and Operational Center (OpCen) Bias and StdDev wind speed estimate values, as validated against collocated aircraft reconnaissance measurements, for HIE intensity estimates using simulated GOES-ABI and current GOES imagery (in m/s).

Some tuning to the HIE scene type and intensity determination schemes may be necessary to account for the improved spatial resolution and resulting scene type samples (i.e. more eye scenes) once a greater sample of higher-temporal data is available (i.e. after GOES-R launch).. In addition, the increased temporal image frequency between the current GOES and GOES-ABI may require some modification to the time-averaging scheme and application of time-base rules. Improvements based upon this proxy data set could be attempted for future HIE algorithm deliveries prior to GOES-R launch, but a more complete proxy dataset analysis would be desired before any significant algorithm modifications are conducted in order to fully assess the impact of the GOES-R ABI data on the HIE algorithm.

4.2 MSG SEVIRI and Simulated GOES-R ABI data validation study

4.2.1 Simulated Input Data Sets

A second validation test of the HIE algorithm (Version 5) using simulated and GOES-R ABI-like data was conducted using data collected and distributed by CIRA. This data set consisted of collocated infrared window channel imagery from the Meteosat Second Generation (MSG) SEVIRI instrument (10.8 μm) and simulated GOES-R ABI (SABI) data (10.3 μm) derived from the MSG data using ABI channel coefficients supplied by the Joint Center for Data Assimilation. Both data sets were presented at 3 km spatial resolution at a 15 minute temporal resolution. Statistical comparisons regarding the spatial aspect of the data sets are expected to be minimal due to the equal resolution between the two data sets. Instead the study will focus on the impact of the improved 15-minute resolution with the ABI data over the 30-minute resolution available with the current GOES imagery.

Data was collected for a total of seven tropical cyclones, during the 2006 to 2008 tropical cyclone seasons, in the eastern and central North Atlantic Ocean basin within the viewing region of the MSG satellite, as listed in Table 16. A total of 5764 images were supplied by CIRA for the seven storms in the validation study after manual quality control measures were taken (visual inspection of each individual image in the data set).

Storm Name & Year	Scenes	Start Date/Time	End Date/Time
Gordon 2006	953	10 Sept / 1800 UTC	20 Sept / 1930 UTC
Helene 2006	1046	12 Sept / 1200 UTC	23 Sept / 1345 UTC
Dean 2006	452	13 Aug / 0600 UTC	17 Aug / 2345 UTC
Karen 2007	422	24 Sept / 0000 UTC	28 Sept / 1200 UTC
Ike 2008	845	30 Aug / 0600 UTC	8 Sept / 1200 UYC
Bertha 2008	1808	30 June / 0600 UTC	20 July / 0645 UTC
Omar 2008	238	16 Oct / 0000 UTC	18 Oct / 1200 UTC
Total	5764		

Table 16: 2006 – 2008 MSG/SABI TC validation case study sample information.

4.2.2 Output from Simulated Input Data Sets

The HIE estimates produced for each storm and data type were initialized using the NHC Best Track intensity information interpolated to the Start Date/Time for the storm being analyzed. Statistical comparisons were homogeneous between all data sets and image times available and were compared to the NHC Best Track wind speed estimates interpolated to the image times being compared (since aircraft reconnaissance data is either not available or severely limited for each storm in the data sample).

The NHC Best Track data was obtained for each storm from the NHC FTP site after a thorough post-storm analysis has been performed to derive the best 6-hour resolution intensity trend of each storm in the data sample. Wind speeds for the HIE estimates were derived from the CI# estimates using the conversion table listed in Table 6.

In general, more error is expected with this data sample due to the fact that many of the individual TC views obtained with the MSG satellite (and derived SABI data) at a high viewing angle (50 or more degrees from nadir). High viewing angles may limit the ability of the satellite to view the structure of the eye region properly, thus affecting the eye temperature value obtained in the imagery. This could affect the intensity estimates from the HIE during eye scene types since the eye temperature could sampling the eye wall instead of the ocean surface. In addition, high viewing angles will also affect various measurement values obtained by the HIE, such as the shear distance values, the amount of cloud curvature with the curved band scene types, and possibly the areal coverage of the CDO region being measured.

4.2.3 Precision and Accuracy Estimates

A total of 2453 homogeneous estimates were found between the 15 and 30 minute temporal resolution data samples for the MSG and SABI data sets. Table 17 displays the statistical measurements for accuracy and precision, absolute bias and standard deviation respectively, for the Version 5 HIE wind speed (in m/s) estimates derived from the CI# and compared to the interpolated NHC Best Track intensity estimate information.

Units m/s	Bias	Accuracy	Precision	Num
SABI – 30 min	1.46	1.46	8.93	2453
SABI – 15 min	1.89	1.89	8.97	2453
MSG – 30 min	1.05	1.05	8.67	2453
MSG – 15 min	1.48	1.48	8.66	2453

Table 17: Comparisons between SABI and MSG data samples for 15 and 30 minutes temporal resolution satellite data availability. Accuracy and Precision are provided by the absolute bias (ABias) and Standard Deviation (StdDev) for the total sample size (Num).

In general, the accuracy measurements were well under the 6.5 m/s threshold for the HIE algorithm, while the precision values were near the threshold 8.0 m/s threshold despite the poor viewing angles available for much of the data sample. Over 60% of the imagery sampled for the comparisons in Table 17 had viewing angles greater than or equal to 50 degrees, and 17.6% had viewing angles greater than 60 degrees.

Overall, MSG data demonstrates slightly better accuracy precision than the SABI data. This is consistent with the results obtained in section 4.1.3 comparing current satellite data (GOES) to simulated ABI data MODIS/AVHRR) even though that data sample was significantly smaller.

Comparing the different temporal resolution data sets, no improvement in either precision or accuracy was obtained with the 15-minute data over the 30-minute data for both the MSG and SABI data samples.

Again, it should be noted that the results from this experiment should not be highlighted despite the large number of data matches between the HIE estimates and the "ground truth" NHC Best Track data. Not only are there a large number of high viewing angle cases, but in general the NHC Best Track estimates are typically on the conservative side, which could lead to larger errors and biases than would be obtained with in situ aircraft reconnaissance measurements of intensity.

4.3 Validation study using current GOES imagery

4.3.1 Input Comparison Data Sets

Ten tropical storms of varying intensity between 2004 and 2008 were analyzed using the HIE (Version 5) using current GOES infrared window (10.7µm) imagery. The ten storms in the data sample are listed by the year in which they occurred:

- 2004: Hurricane Alex, Hurricane Charley
- 2005: Tropical Storm Arlene, Hurricane Dennis, Hurricane Katrina, Hurricane Rita
- 2006: Hurricane Ernesto
- 2007: Hurricane Dean
- 2008: Hurricane Dolly, Hurricane Gustav

This study will serve as a baseline for the algorithm accuracy since only current goes imagery at 4 km spatial resolution and 30 minute temporal resolution is used without any MW eye score input.

4.3.2 Output from current GOES data sets

Aircraft reconnaissance in situ measurements of intensity will be used to validate the accuracy of the HIE intensity estimates. Measurements of TC central MSLP will be converted to wind speed (in m/s) and compared to the HIE estimates within one hours of the reconnaissance data. Statistical measurements of accuracy and precision will be compared for the intensity estimates derived from the Version 5 HIE algorithm and for the corresponding ensemble (of SAB and NHC estimates when both are available) Operational Center (OpCen) estimates (within 30 minutes of the HIE intensity estimate).

4.3.3 Precision and Accuracy Estimates

Table 18 displays the statistical comparisons between the HIE (Version 5 – 100% delivery) and the OpCen estimates of intensity in terms of the AWG definitions of accuracy (absolute bias (ABias)) and precision (standard deviation (StdDev)) along with the bias of each method for the 529 homogeneous sample comparisons.

As with the other studies listed in this document, intensity differences are given in terms of HIE intensity estimates or OpCen estimates minus aircraft reconnaissance, so a positive/negative bias indicates an over/underestimate of intensity by the HIE or OpCen in comparison to the aircraft reconnaissance in situ measurement.

Units m/s	Bias	Accuracy	Precision	Num
HIE Version 5 (100%)	-5.91	5.91	7.60	529
OpCen	-2.42	2.42	6.27	529

Table 18: Comparisons between the HIE (Version 5) and the Operational Forecast Center (OpCen) intensity estimates versus aircraft reconnaissance measurements of intensity for the 10 case validation TC sample.

4.4 Passive Microwave (PMW) Eye Score validation study

4.4.1 Input Data Sets

In order to determine the impact of use of the PMW Eye Score (described in Section 3.4.2.12) values on the HIE algorithm, a final validation data set was collected containing 40 storms between 2005 and 2008 where the PMW Eye Score values were available. This study will also assess the impact of all changes between Version 3, the 80% code delivery and version used in all previous validation tests, and Version 5, the 100% code delivery version. The inclusion of the PMW values within the HIE is the last major upgrade to the HIE algorithm prior to 100% delivery to AIT.

The scope of this study is to examine a large data set to properly assess the impact of the PMW Eye Score data on the HIE. Previous impact study data sets were either too small or had significant limitations, as discussed in the individual sections above, to properly implement and assess the use the PMW Eye Score values on the individual data sets. Utilization of current GOES data, available at the time of each TC in the study, will provide an upper boundary for the statistical validation of the HIE using a large data sample. Statistical improvements using the simulated ABI data from the previous validation studies can be implied to this data sample to estimate the statistical accuracy which could be obtained with the improved ABI imagery.

4.4.2 Output from PMW Eye Score validation study

Aircraft reconnaissance in situ measurements of intensity will be used to validate the accuracy of the HIE intensity estimates. Measurements of TC central MSLP will be converted to wind speed (in m/s) and compared to the HIE estimates within one hours of the reconnaissance data. Statistical measurements of accuracy and precision will be compared for the intensity estimates derived from the Version 3 and Version 5 HIE algorithms.

4.4.3 Precision and Accuracy Estimates

Table 19 displays the statistical comparisons between the two HIE versions used for this validation study in terms of the AWG definitions of accuracy (absolute bias (ABias)) and precision (standard deviation (StdDev)) along with the bias of each method for the 744 homogeneous sample comparisons.

As with the other studies listed in this document, intensity differences are given in terms of HIE intensity estimates minus aircraft reconnaissance, so a positive/negative bias indicates an over/underestimate of intensity by the HIE in comparison to the aircraft reconnaissance in situ measurement.

Units m/s	Bias	Accuracy	Precision	Num
HIE - Version 3	-2.48	2.48	7.22	744
HIE - Version 5	-2.45	2.45	6.75	744

Table 19: Comparisons between the Version 3 and Version 5 HIE algorithms versus aircraft reconnaissance measurements of intensity for the 40 case TC sample from 2005 to 2008.

4.5 Error Budget

As shown in the statistical results presented in the previous sections, the accuracy of the HIE algorithm (Versions 3 and 5 of the algorithm) using various current and simulated GOES-R ABI data sets are close to or exceeding the AWG measurement accuracy requirements of 6.5 m/s for accuracy and 8.0 m/s for precision. In general, the requirement for accuracy has been achieved for each validation study. The precision requirement was met in the studies presented in Section 4.1, 4.3, and 4.4, however it was not met in the study presented in Section 4.2. Taking into account the limitations presented in the Section 4.2 study with high-viewing angle issues and use of non-in situ "ground truth" data the HIE algorithm has met and exceeded the required specifications thresholds for accuracy and precision, and it is fully expected to meet and exceed the AWG specified levels of accuracy and precision required for the hurricane intensity algorithm.

The results from the validation study presented in Sections 4.3 and 4.4 provide the largest statistical data samples versus aircraft in situ measurements of intensity and show the impact of the differences between Version 3 (80% delivery) and Version 5 (100% delivery). The results from these validation studies not only demonstrates the improved accuracy of the HIE algorithm between the two versions, but also in the utilization of the PMW Eye Score data. These results clearly demonstrates the ability of the HIE algorithm to exceed the spec requirements for the HIE algorithm despite using current GOES imagery instead of simulated GOES-R ABI data. From these studies, it has been shown that the Version 5 HIE algorithm is ready for GOES-R launch.

5 PRACTICAL CONSIDERATIONS

5.1 Numerical Computation Considerations

The HIE is a self-contained algorithm which does not rely on any external processes to be completed prior to the start of the intensity estimation process. An external passive microwave eye score value (Section 3.4.2.12) can be derived and passed into the HIE upon initiation of the routine execution, but the HIE does not depend on this value in order to run successfully.

5.2 Programming and Procedural Considerations

The HIE has been checked for memory leaks and other programming errors using the open source software "Valgrind" which is freely available under the GNU General Public License by the Valgrind Developers Group at valgrind.org.

5.3 Quality Assessment and Diagnostics

The following procedures are recommended for diagnosing the performance of the HIE as the algorithm is being executed in real-time.

- Monitor the automated HIE output to check that the scene types are matching what the forecaster would expect to obtain if manually performing the SDT. For example, if the user expects to see and eye scene but obtains something other than an eye, the automated storm center position value should be investigated.
- Assess the accuracy of the automated storm center position determination process, especially when a forecast interpolation is being performed. If the center is not in the region were the HIE user would expect the storm center to be, make sure the forecast being utilized is the latest available. If it is not available, the use of an official forecast from another TCFC may be considered for use.
- If an individual temperature value is found to be inconsistent with surrounding values in the history file, check the image for data dropouts and/or bad pixels.
- Periodically cross-check the HIE intensity estimates with other available estimates (satellite or aircraft-based) from in-house sources.

5.4 Exception Handling

The HIE performs validity checks for the various input products and internally determined values during the execution of the HIE. Various error and diagnostic messages are produced during the execution of the HIE and output at the end of the processing cycle. These messages can be used by the user to determine why a particular HIE run failed or how the HIE obtained a certain intensity estimate value. See Section 3.4.3.4 for the tables that display the various HIE Error and Diagnostic codes in terms of the internal HIE code values and the corresponding error values in a bit/byte format desired for the AIT Framework.

5.5 Algorithm Validation

Validation of the HIE tropical cyclone intensity estimates requires collocated measurements of reference (“ground truth”) tropical cyclone surface maximum wind speed. These measurements, as discussed in Section 4, are typically obtained from in situ reconnaissance aircraft using dropsondes to measure the at or near surface wind speeds within the tropical cyclone eyewall. Also, measurements of minimum MSLP within the storm environment can be used and converted to wind speed using either the standard pressure/wind speed relationship presented in Section 3.4.2.5 or with newer/experimental methods such as discussed in Section 3.3.2.5. Whichever method is used to convert the MSLP to wind speed should be consistent with the methodology used in the derivation of the HIE wind speed estimate values.

If direct in situ aircraft measurements of the TC intensity are not available, official “Best Track” estimates of intensity can be used as a proxy for the in situ data. The official Best Track is derived for every storm identified by the TCFC responsible for the basin in question. The Best Track values are derived using all available methods of measuring and/or estimating TC intensity, including in situ aircraft measurements (when available), satellite-based estimation algorithms (including the ADT and SDT), surface reports from fixed land and ocean-based observing stations, ship reports, and scatterometer measurements. When aircraft measurements are not available, the Best Track data quality is obviously degraded and will tend to more conservative estimates, but the information can still be used as a method of validation, especially in ocean basins outside of the Northeastern Atlantic Ocean where aircraft reconnaissance does not exist.

During the pre-launch phase of the GOES-R program, the validation activities are aimed at characterizing the precision and accuracy of the HIE algorithm to make sure it meets the AWG specifications for algorithm quality. Also, comparisons between the offline and framework versions of the algorithm will be conducted to characterize any differences between the versions based upon the implementation environments. These data sets have been described in Section 4.1.

Post-launch verification will be conducted in much the same fashion as the pre-launch verification. Characterizations of the algorithm precision and accuracy will be derived preferably using in-situ measurements of TC intensity from aircraft reconnaissance, but TCFC Best Track estimates can be used in regions where no in situ data is available.

Validation methodologies and tools developed and tested during the pre-launch phase will be automated and applied as necessary. More specific details on the HIE product validation activities can be found in the Product Validation document for the HIE algorithm.

6 ASSUMPTIONS AND LIMITATIONS

The following sections describe the current limitations of the HIE code, and assumptions when utilizing the code operationally.

6.1 Performance

When the HIE is utilized in the GOES-R operational environment, the following will be assumed when the algorithm is executed and its performance is analyzed.

1. The HIE will be executed in a completely automated fashion without any user interaction. Official TCFC forecasts are updated every 6 hours, immediately available upon official public release, and are properly formatted. This information will provide initial guess positions for the HIE auto-positioning logic.
2. Satellite imagery is available minimally every 30 minutes. Utilization of imagery at a lower temporal resolution will negatively impact the performance of the HIE, especially the time-averaging routine and time-dependent intensity change rules. Use of higher temporal resolution data has not yet been specifically quantified but has been shown in Section 4.2 to marginally improve the accuracy of the algorithm. This experiment was limited to a small sample of data, so any information gleaned from this study should be investigated further with a larger sample.
3. The HIE is executed only over ocean surfaces. The HIE should not (and will not in the default execution) be operated over land surfaces.

6.2 Assumed Sensor Performance

The following satellite sensor characteristics will be assumed when utilizing the HIE.

1. All satellite channel calibration will be correct and uniform from image to image. Correct intensity estimations are reliant upon satellite cloud top temperature information.
2. Satellite navigation errors are within specifications as defined in the Performance and Operational Requirements Document PORD. Correct satellite navigation is imperative for automated storm center determination and TC center position placement following the forecast interpolation process.
3. The ABI instrument will perform within the noise specifications defined in the PORD. The HIE can detect certain discrepancies in satellite data accuracy/availability, however large regions of invalid data will cause failure of the algorithm.

6.3 Pre-Planned Product Improvements

The HIE is constantly undergoing modifications to improve the accuracy of the intensity estimates determined by the algorithm. Issues or biases are often discovered during the utilization of the HIE algorithm during a specific TC season. Many of these are minor improvements; however some are more drastic, requiring a more in-depth research and development process. Below are issues that are currently being investigated to improve aspects of the HIE operation and performance as noted by CIMSS scientists and/or operational TCFC forecasters who utilize the HIE.

6.3.1 Improvement 1

Intensity estimates during TC land interaction (leading up to landfall and after re-emergence). Further research to optimize the HIE performance in these situations should be conducted

6.3.2 Improvement 2

Improvement of the intensity estimates produced during weak TC situations prior to formation of an eye. This situation is commonly referred to as the "central dense overcast" (CDO) case when a uniform (temperature) cloud shield masks a possibly forming eye underneath. Additional ancillary information can further improve the HIE accuracy. These include:

a. Utilization of an externally-derived "eye score" parameter obtained from coincident passive microwave imagery from polar orbiting satellites. This technique is currently being investigated at CIMSS in an experimental mode. Further modifications and complete evaluation of the technique will be performed before incorporation into the GOES-R HIE. (Update: This has been incorporated into the HIE algorithm and presented in Sections 3.4.2.12 and 4.4)

b. Utilization of an infrared and water vapor channel differencing scheme to detect and measure intense convection near the storm center. This method can be used to aid in the storm center determination process during the CDO cases when the IR image proves to be inadequate for identifying a storm center. Initial research studies are encouraging, with more investigation needed before integration into the GOES-R HIE.

7 REFERENCES

Courtney, J., and J.A. Knaff, 2009: Adapting the Knaff and Zehr wind-pressure relationship for operational use in Tropical Cyclone Warning Centres, *Australian Meteor. and Ocean. Journal*, **58**, 167-179.

Dvorak, V., 1975: Tropical cyclone intensity analysis and forecasting from satellite imagery. *Mon. Wea. Rev.*, **103**, 420-430.

Dvorak, V., 1984: Tropical cyclone intensity analysis using satellite data. NOAA Tech. Rep. NESDIS 11, 47 pp. [Available from NOAA/NESDIS, 5200 Auth Rd., Washington, DC 20233.]

Knaff, J.A., and R.M. Zehr, 2007: Reexamination of Tropical Cyclone Wind-Pressure Relationships. *Wea. Forecasting*, **22**, 71-88.

Kossin, J.P. and C.S. Velden, 2003: A pronounced bias is tropical cyclone intensity estimation based on the Dvorak Technique, *Mon. Wea. Rev.*, **132,** 165-173.

Olander, T.L., and C.S. Velden, 2007: The Advanced Dvorak Technique (ADT) – Continued development of an objective scheme to estimate tropical cyclone intensity using geostationary infrared satellite data. *Wea. Forecasting*., **22**, 287-298.

Olander, T.L., C.S. Velden, and J.P. Kossin, 2004: The Advanced Objective Dvorak Technique (AODT) – Latest upgrades and future directions. *26th Conference on Hurricanes and Tropical Meteorology*, Miami, FL, Amer. Meteor. Soc., 294-295.

Olander, T.L., C.S. Velden, and M.A. Turk, 2002: Development of the Advanced Objective Dvorak Technique (AODT) – Current progress and future directions. *25th Conference on Hurricanes and Tropical Meteorology*, San Diego, CA, Amer. Meteor. Soc., 585-586.

Shewchuck, J.D. and R.C. Weir, 1980: An evaluation of the Dvorak technique for estimating tropical cyclone intensity from satellite imagery. NOCC/JTWC 80-2, USNOCC, JTWC, Comnavmarinas, Box 17, San Francisco, CA, 96630, 25 pp.

Velden, C.S, T.L. Olander and R. Zehr, 1998: Development of an objective scheme to estimate tropical cyclone intensity from digital geostationary satellite infrared imagery. *Wea. Forecasting*, **13**, 172-186.

Wimmers A., and C. Velden, 2004: Satellite-based center-fixing of tropical cyclones: new automated approaches. *26th Conference on Hurricanes and Tropical Meteorology*, Miami, FL, Amer. Meteor. Soc., 82-83.

Zehr, R., 1989: Improving objective satellite estimates of tropical cyclone intensity. *Extended Abstracts, 18th Conf. on Hurricanes and Tropical Meteorology*, San Diego, CA, Amer. Meteor. Soc., J25-J28.

8 Appendix

8.1 netCDF History File header listing

```
netcdf Winds_ADT_KATRINA05 {
dimensions:
   Record = 247 ;
   chid = 40 ;
variables:
   float Latitude(Record) ;
      Latitude:long_name = "Latitude" ;
      Latitude:units = "degrees_north" ;
      Latitude:comments = "Pixel latitude (degree)" ;
      Latitude:_FillValue = -999.f ;
      Latitude:valid_range = -90.f, 90.f ;
   float Longitude(Record) ;
      Longitude:long_name = "Longitude" ;
      Longitude:units = "degrees_east" ;
      Longitude:comments = "Pixel Longitude (degree)" ;
      Longitude:_FillValue = -999.f ;
      Longitude:valid_range = -180.f, 180.f ;
   int Time(Record) ;
      Time:long_name = "time of measurement" ;
      Time:units = "secs since 1970-01-01 00:00:00" ;
   float TrawO(Record) ;
      TrawO:long_name = "Raw T# (unadjusted value)" ;
      TrawO:units = "m/s" ;
      TrawO:coordinates = "Time Latitude Longitude" ;
      TrawO:_FillValue = 0.f ;
      TrawO:valid_range = 1.f, 8.5f ;
   float Traw(Record) ;
      Traw:long_name = "Raw T# (adjusted value, after constraint limits applied)" ;
      Traw:units = "m/s" ;
      Traw:coordinates = "Time Latitude Longitude" ;
      Traw:_FillValue = 0.f ;
      Traw:valid_range = 1.f, 8.5f ;
   float Tfinal(Record) ;
      Tfinal:long_name = "Final T# (3 hour averaged)" ;
      Tfinal:units = "m/s" ;
```

```
        Tfinal:coordinates = "Time Latitude Longitude" ;
        Tfinal:_FillValue = 0.f ;
        Tfinal:valid_range = 1.f, 8.5f ;
    float CI(Record) ;
        CI:long_name = "cyclone intensity value" ;
        CI:units = "m/s" ;
        CI:coordinates = "Time Latitude Longitude" ;
        CI:_FillValue = 0.f ;
        CI:valid_range = 1.f, 8.5f ;
    float WndSpeed(Record) ;
        WndSpeed:long_name = "Wind Speed" ;
        WndSpeed:units = "m/s" ;
        WndSpeed:coordinates = "Time Latitude Longitude" ;
        WndSpeed:_FillValue = 0.f ;
        WndSpeed:valid_range = 12.f, 103.f ;
    float eyet(Record) ;
        eyet:long_name = "Eye Region Temperature" ;
        eyet:units = "deg C" ;
        eyet:coordinates = "Time Latitude Longitude" ;
        eyet:_FillValue = 99.99f ;
        eyet:valid_range = -100.f, 30.f ;
    float cloudt(Record) ;
        cloudt:long_name = "Cloud Region Temperature" ;
        cloudt:units = "deg C" ;
        cloudt:coordinates = "Time Latitude Longitude" ;
        cloudt:_FillValue = 99.99f ;
        cloudt:valid_range = -100.f, 30.f ;
    float cloudt2(Record) ;
        cloudt2:long_name = "Cloud Region Temperature; Old Value" ;
        cloudt2:units = "deg C" ;
        cloudt2:coordinates = "Time Latitude Longitude" ;
        cloudt2:_FillValue = 99.99f ;
        cloudt2:valid_range = -100.f, 30.f ;
    float cwcloudt(Record) ;
        cwcloudt:long_name = "Cloud Region \"coldest warmest ring\" Temperature" ;
        cwcloudt:units = "deg C" ;
        cwcloudt:coordinates = "Time Latitude Longitude" ;
        cwcloudt:_FillValue = 99.99f ;
        cwcloudt:valid_range = -100.f, 30.f ;
    float eyecdosize(Record) ;
         eyecdosize:long_name = "eyecdosize=Eye or CDO Size value (km; value is
scene dependent)" ;
        eyecdosize:units = "deg C" ;
        eyecdosize:coordinates = "Time Latitude Longitude" ;
        eyecdosize:_FillValue = -99.5f ;
        eyecdosize:valid_range = 0.1f, 999.9f ;
```

```
float eyestdv(Record) ;
   eyestdv:long_name = "Standard deviation of eye region temperature values" ;
   eyestdv:units = "deg C" ;
   eyestdv:coordinates = "Time Latitude Longitude" ;
   eyestdv:_FillValue = 0.f ;
   eyestdv:valid_range = 0.1f, 99.9f ;
float cloudsymave(Record) ;
   cloudsymave:long_name = "Cloud Region symmetry value" ;
   cloudsymave:units = "1" ;
   cloudsymave:coordinates = "Time Latitude Longitude" ;
   cloudsymave:_FillValue = 0.f ;
   cloudsymave:valid_range = 0.1f, 99.9f ;
int sattype(Record) ;
   sattype:long_name = "Internal ADT Satellite ID value" ;
   sattype:units = "1" ;
   sattype:coordinates = "Time Latitude Longitude" ;
   sattype:_FillValue = -999 ;
   sattype:valid_range = 0, 22 ;
int eyescene(Record) ;
   eyescene:long_name = "Eye Scene ID value" ;
   eyescene:units = "1" ;
   eyescene:coordinates = "Time Latitude Longitude" ;
   eyescene:_FillValue = -999 ;
   eyescene:valid_range = 0, 3 ;
int cloudscene(Record) ;
   cloudscene:long_name = "Cloud Scene ID value" ;
   cloudscene:units = "1" ;
   cloudscene:coordinates = "Time Latitude Longitude" ;
   cloudscene:_FillValue = -999 ;
   cloudscene:valid_range = 0, 4 ;
int eyesceneold(Record) ;
   eyesceneold:long_name = "Original Eye Scene ID value(if user override)" ;
   eyesceneold:units = "1" ;
   eyesceneold:coordinates = "Time Latitude Longitude" ;
   eyesceneold:_FillValue = -1 ;
   eyesceneold:valid_range = 0, 3 ;
int cloudsceneold(Record) ;
   cloudsceneold:long_name = "Original Cloud Scene ID value(if user override)" ;
   cloudsceneold:units = "1" ;
   cloudsceneold:coordinates = "Time Latitude Longitude" ;
   cloudsceneold:_FillValue = -1 ;
   cloudsceneold:valid_range = 0, 4 ;
int rule9(Record) ;
   rule9:long_name = "Dvorak Rule 9 Flag value" ;
   rule9:units = "1" ;
   rule9:coordinates = "Time Latitude Longitude" ;
   rule9:_FillValue = -999 ;
   rule9:valid_range = 0, 2 ;
int rule8(Record) ;
   rule8:long_name = "Dvorak Rule 8 Flag value" ;
   rule8:units = "1" ;
   rule8:coordinates = "Time Latitude Longitude" ;
   rule8:_FillValue = -999 ;
   rule8:valid_range = 0, 33 ;
int LBflag(Record) ;
   LBflag:long_name = "Latitude Bias Flag value" ;
   LBflag:units = "1" ;
   LBflag:coordinates = "Time Latitude Longitude" ;
   LBflag:_FillValue = -999 ;
   LBflag:valid_range = 0, 2 ;
int rapiddiss(Record) ;
   rapiddiss:long_name = "Rapid Dissipation Flag" ;
   rapiddiss:units = "1" ;
   rapiddiss:coordinates = "Time Latitude Longitude" ;
   rapiddiss:_FillValue = -999 ;
   rapiddiss:valid_range = 0, 3 ;
int land(Record) ;
   land:long_name = "Land Flag ID value" ;
   land:units = "1" ;
   land:coordinates = "Time Latitude Longitude" ;
   land:_FillValue = 0 ;
   land:valid_range = 1, 2 ;
int eyefft(Record) ;
   eyefft:long_name = "Eye Region Fast Fourier Transform ID value" ;
   eyefft:units = "1" ;
   eyefft:coordinates = "Time Latitude Longitude" ;
   eyefft:_FillValue = -999 ;
   eyefft:valid_range = 0, 13 ;
int cloudfft(Record) ;
   cloudfft:long_name = "Cloud Region Fast Fourier Transform ID value" ;
   cloudfft:units = "1" ;
   cloudfft:coordinates = "Time Latitude Longitude" ;
   cloudfft:_FillValue = -999 ;
   cloudfft:valid_range = 0, 13 ;
int ringcb(Record) ;
   ringcb:long_name = "Curved Band Ring Gray Scale Value" ;
   ringcb:units = "1" ;
   ringcb:coordinates = "Time Latitude Longitude" ;
   ringcb:_FillValue = 0 ;
   ringcb:valid_range = 2, 6 ;
int ringcbval(Record) ;
   ringcbval:long_name = "Curved Band Curvature Amount" ;
   ringcbval:units = "1" ;
   ringcbval:coordinates = "Time Latitude Longitude" ;
   ringcbval:_FillValue = 0 ;
   ringcbval:valid_range = 8, 25 ;
int autopos(Record) ;
   autopos:long_name = "Automated Centering Method Value" ;
   autopos:units = "1" ;
   autopos:coordinates = "Time Latitude Longitude" ;
   autopos:_FillValue = -999 ;
   autopos:valid_range = 0, 6 ;
int cwring(Record) ;
   cwring:long_name = "time latitude longitude" ;
   cwring:units = "km" ;
   cwring:coordinates = "Time Latitude Longitude" ;
   cwring:_FillValue = 0 ;
   cwring:valid_range = 24, 136 ;
float CIadjp(Record) ;
   CIadjp:long_name = "Latitude Bias MSLP adjustment" ;
   CIadjp:units = "hPa" ;
   CIadjp:coordinates = "Time Latitude Longitude" ;
   CIadjp:_FillValue = -999.f ;
   CIadjp:valid_range = -20.f, 20.f ;
float rmw(Record) ;
   rmw:long_name = "Radius of Maximum Wind" ;
   rmw:units = "km" ;
   rmw:coordinates = "Time Latitude Longitude" ;
   rmw:_FillValue = -99.5f ;
   rmw:valid_range = 2.f, 136.f ;
float MWScore(Record) ;
   MWScore:long_name = "Microwave Eye Score value" ;
   MWScore:units = "1" ;
   MWScore:coordinates = "Time Latitude Longitude" ;
   MWScore:_FillValue = -99.f ;
   MWScore:valid_range = 0.f, 100.f ;
float R34(Record) ;
   R34:long_name = "Radius of gale (34knt) winds" ;
   R34:units = "nautical miles" ;
   R34:coordinates = "Time Latitude Longitude" ;
   R34:_FillValue = -99.f ;
   R34:valid_range = 0.f, 999.f ;
float MSLPenv(Record) ;
   MSLPenv:long_name = "Knaff/Zehr environmental MSLP value" ;
   MSLPenv:units = "hpa" ;
   MSLPenv:coordinates = "Time Latitude Longitude" ;
   MSLPenv:_FillValue = -999.f ;
   MSLPenv:valid_range = 900.f, 1050.f ;
```

```
    int StormNum(Record) ;
        StormNum:long_name = "Storm Number" ;
        StormNum:units = "1" ;
        StormNum:coordinates = "Time Latitude Longitude" ;
        StormNum:_FillValue = -999 ;
        StormNum:valid_range = 1, 50 ;
    char StormName(Record, chid) ;
        StormName:long_name = "StormName" ;
        StormName:units = "1" ;
        StormName:coordinates = "Time Latitude Longitude" ;
    char BasinID(Record, chid) ;
        BasinID:long_name = "Basin ID" ;
        BasinID:units = "1" ;
        BasinID:coordinates = "Time Latitude Longitude" ;
    int NumMeasurements(Record) ;
        NumMeasurements:long_name = "Number of hurricane intensity measurements
for the storm" ;
        NumMeasurements:units = "1" ;
        NumMeasurements:coordinates = "Time Latitude Longitude" ;
        NumMeasurements:_FillValue = -999 ;
        NumMeasurements:valid_range = 1, 1000 ;
    float CIDiff(Record) ;
        CIDiff:long_name = "Difference of hurricane intensity between current image
and prior measurment" ;
        CIDiff:units = "m/s" ;
        CIDiff:coordinates = "Time Latitude Longitude" ;
        CIDiff:_FillValue = -999.f ;
        CIDiff:valid_range = 0.f, 8.5f ;

// global attributes:
        :History = "Thu Jun 28 17:47:56 2012" ;
        :Image_Date = "2005241" ;
        :Image_Time = "1815_00" ;
        :Conventions = "CF-1.4" ;
        :Title = "AIT Wind ADT" ;
        :Source = "AIT_Framework" ;
```

8.2 Scene Type Determination Pseudo-Code

A description of the scene type determination process using pseudo-code is provided below:

```
CloudBDCategoryF = AER Table3-5 BD enhancement Category for Cloud Temperature
EyeBDCategoryF = AER Table3-5 BD enhancement Category for Eye Temperature
CloudCWBDCategoryF = AER Table3-5 BD enhancement Category for Cloud CW Temperature
CloudBDCategoryI = AER Table3-5 BD enhancement Category for Cloud Temperature
```

```
    EyeBDCategoryI = AER Table3-5 BD enhancement Category for Eye Temperature
    CloudCWBDCategoryI = AER Table3-5 BD enhancement Category for Cloud CW Temperature
```

Note with above: “F” indicates Float Value, while “I” indicate Integer value. This is as follows; if the temperature is, for example, -73C, the “Float” BD category value will be 6.5, while the “Integer” will be 6.

```
    BDCatDifferenceI = MAX(0,MAX(CloudBDCategoryI,CloudCWBDCategoryI)-
EyeBDCategoryI)
    EyeCloudTempDiff = Eye Region temperature – Cloud Region temperature difference
    CloudTempDiff = Cloud region temperature – Cloud region “CW” Temperature
    EyeCloudCWBDCatDiffF = EyeBDCategoryF – CloudCWBDCategoryF
    EyeCloudBDCatDiffI = CloudBDCategoryI – EyeBDCategoryI
    EyeCloudBDCatDiffF = CloudBDCategoryF – EyeBDCategoryF
    CloudBDCatDiffI = CloudCWBDCategoryI – CloudBDCategoryI
    CloudBDCatDiffF = CloudBDCategoryF – CloudBDCWCategoryF
    CloudCWBDCatDiffF = CloudBDCategoryF - CloudCWBDCategoryF
    EyeCloudTempDiff2 = Eye Region temperature – MIN(Cloud region temperature,Cloud region
“CW” Tempearure)
    EyeFFTValue = Fast Fourier Transform analysis harmonic value for Eye Region
    CloudFFTValue = Fast Fourier Transform analysis harmonic value for Cloud Region
```

```
Determine Eye Region Scene Score
    EyeValueC = 0.0
    EyeValueE = 0.0
    EyeValueA = 1.0 – (EyeFFTValue – 2) * 0.1)
    EyeValueB = EyeBDCategoryF * 0.5
    If (EyeValueB > 10.0)
            EyeValueC  =0.50
    Else
            EyeValueC – 0.0
    End
    EyeValueD = (EyeCloudBDCatDiffF * 0.25) + (EyeCloudCWBDCatDiff * 0.50)
    If (found record T-12hours) and (previous scene is not an eye) and (storm intensity has exceeded
5.0 during history) then EyeValueC = EyeValueC + 0.25
    If (FinalT#Value T-12hours <= 4.5) then
            EyeValueE = MAX(-1.0,(FinalT#Value T-12hours – 4.5))
    End
    If (Previous Rule9 Value>0) and (previous FinalT# value < 4.0) then
            EyeValueE = EyeValueE – 0.5
    End
    EyeSceneScore = EyeValueA + EyeValueB + EyeValueC + EyeValueD + EyeValueE
    If ( EyeSceneScore >= 0.5) then
            Scene is an EYE
    Else
            Scene is NOT AN EYE
    End
    Determine Eye Size using Radius of Max Wind program (which outputs radius of max wind and
eye size)
    If (Eye Size > Large Eye Size Threshold) Scene is a LARGE EYE
```

```
Determine Cloud Scene Score
    CloudValueC = 0.0
    CloudValueD = 0.5
    CloudValueE = 0.0
    CloudValueA = CloudCWBDCategoryF * 0.25
    CloudValueB = CloudBDCategoryF * 0.25
    If (CloudFFTValue <= 2) then
```

```
        CloudValueC = MIN(1.50,(CloudValueA * 0.25))
    End
    If (Previous cloud scene was SHEAR or CURVED BAND) CloudValueD = -0.50
    If (CloudCWBDCategoryF > 2.0) then
        If (Eye Scene was found above)
            CloudValueE = MIN(1.0,(Previous FinalT# T-12hours – 2.5))
        End
        If (Previous FinalT# T-12hours >= 3.5) then
            CloudValueE = CloudValueE + 1.00
        End
    End
    If (found record T-12hours) and (Found an eye scene previously in history of storm) then
        CloudValueE = CloudValueE + 1.25
    End
    CloudSceneScore = ClouDValueA + CloudValueB + CloudValueC + CloudValueD + CloudValueE
    If (CloudSceneScore < 0.0) Scene is SHEAR
    If (CloudSceneScore >= 0.0) Scene is CURVED BAND
    If (CloudSceneScore >= 1.0) then
        Scene is CURVED BAND
        If (EyeCloudTempDiff2<0.0) and (Cloud Symmetry Value>40.0)
            Scene is IRREGULAR CDO
        End
    End
    If (CloudSceneScore >2.0 and <3.0) then
        Cloud scene is CURVED BAND
        If (EyeCloudTempDiff2 <0.0) and (Cloud Symmetry Value > 30.0)
            Cloud scene is IRREGULAR CDO
        End
        If (CloudCWBDCategoryI >= 3) then
            If (CloudBDCatDiffI > 0) and (CloudTempDiff < -8.0) then
                Cloud Scene is CURVED BAND (look at Black/White BD gray shade)
            End
            If (Eye Scene found above is an EYE (not a large eye)) OR
              ((EyeBDCategory > 1.0) and (EyeCloudBDCatDiffF > 2.0)) then
                Cloud Scene is NOT a CURVED BAND and is an EYE
            End
            If ((CloudBDCatDiffF <= 0.0) and (EyeCloudCWBDCatDiffF < 1.0)) then
                Cloud Scene is a CDO
            End
        End
    End
    If (CloudSceneScore >= 3.0) then
        Cloud scene is a CDO
        If (CloudBDCatDiffI < 0) and (CloudTempDiff > 8.0) and (Cloud symmetry Value >30.0)
            Cloud Scene is an IRREGULAR CDO
        End
    End
    If (CloudTemperature < Cloud CW Temperature) and (Cloud CW Temperature < Eye Temperature) check for Embedded Center
    If (check for embedded center) and (Cloud scene is not a SHEAR or CURVED BAND above) then
        Determine log spiral wrap amount
        If (log spiral amount >=8 and < 20)) Cloud Scene is an EMBEDDED CENTER
    End
```

```
If (Cloud Scene is a CURVED BAND or SHEAR above) then
    If (Cloud Scene is a SHEAR) then
        Determine Shear Distance
    Else
        If (search for CURVED BAND in gray scale) then
            Start curved band calculation with BD Enhancement Range of 4 (Light Gray)
            Determine log spiral amount
            If (log spiral amount >=8) then
                Select gray scale for Curved Band and set amount
                If (Curved Band amount > 25 and BD Enhancement Range = 4)
                    Try Curved Band analysis using Black/White gray scale
                End
            Else
                Try warmer gray scale down to Enhancement Value 2
                If (BD Enhancement Range = 2 and Curved Band amount < 8)
                    Scene Type is SHEAR
                    Determine Shear Distance
                Else
                    Select Curved Band gray scale and set amount
                End
            End
        End
        If (Check for Curved Band analysis in Black/White gray scale) then
            Determine curved band amount for Black/White gray scales
            If (curved band amount >=9 and <=25) then
                Select Curved Band gray scale and set amount
            Else
                Scene type is CDO or EMBEDDED CENTER
                Determine CDO size
            End
        End
    End
Else
    If (Scene is CDO, Embedded Center, or Irregular CDO) then
        Determine CDO size
        Perform Pinhole Eye check
        If ((EyeSceneScore>-0.25 and < 1.50) AND (EyeCloudBDCatDiffI>=2) AND
          (EyeFFTValue<=2) AND (CloudCWBDCatDiffF>6.0) AND
          (CloudScene is either CDO or Embedded Center) AND (CloudFFTValue <=4)
          AND (Previous FinalT# value T-12hours >= 3.5) then
            Scene type is Pinhole Eye
        End
    Else
        Determine Eye Size
    End
End
```

8.3 Radius of Maximum Wind Determination code

The RMW routine is a regression-based, iterative routine developed by Jim Kossin and others here at CIMSS. The code is provided below.

```
{
 int Ret_ID;                    /* return ID variable */
 int XDirMinimum;                  /* X direction minumum value */
 int XDirMaximum;                  /* X direction maxumum value */
 int YDirMinimum;                  /* Y direction minumum value */
 int YDirMaximum;                  /* Y direction maxumum value */
 int CenterPointXDir;              /* X direction array center point */
 int CenterPointYDir;              /* Y direction array center point */
 int XInc;                     /* X direction incrememt */
 int YInc;                     /* Y direction incrememt */
 int Iteration;                 /* iteration number */
 int XDirIterationStoredMinimum;      /* X direction stored min value */
 int YDirIterationStoredMinimum;      /* Y direction stored max value */
 int XDirIterationStoredMaximum;      /* X direction stored min value */
 int YDirIterationStoredMaximum;      /* Y direction stored max value */
 float DistanceValueX1;             /* X direction distance value */
 float DistanceValueX2;             /* X direction distance value */
 float DistanceValueY1;             /* Y direction distance value */
 float DistanceValueY2;             /* Y direction distance value */
 float AveragedDistance;             /* average distance value */
 float LatitudeValue;              /* latitude value */
 float LongitudeValue;              /* longitude value */
 float CenterPointLatitude;          /* center point latitude value */
 float CenterPointLongitude;          /* center point longitude value */
 float AngleReturnValue;             /* angle value between two points */
 float CriticalTemperatureDegK=228.0;  /* critical temperature value deg K */
 float ThresholdWarmCloudTemperatureDegK=223.0; /* threshold value deg K */

 Ret_ID=0;
 /* calculate cursorx/cursory from numx/numy... values should be 0.5*numx/y */
 CenterPointXDir=IRDataGrid_Global->numx/2;
 CenterPointYDir=IRDataGrid_Global->numy/2;
 XDirMaximum=A_MIN(IRDataGrid_Global->numx,CenterPointXDir+320);
 XDirMinimum=A_MAX(0,CenterPointXDir-320);
 YDirMaximum=A_MIN(IRDataGrid_Global->numy,CenterPointYDir+240);
 YDirMinimum=A_MAX(0,CenterPointYDir-240);

 if(HistoryCurrentPointer_Global->IR.cloudt>=
   ThresholdWarmCloudTemperatureDegK) {
   CriticalTemperatureDegK=(HistoryCurrentPointer_Global->IR.eyet+
            (2.0*HistoryCurrentPointer_Global->IR.cloudt))/3.0;
 }

 /* *RadiusMaxWind_Return=-99.9; */
 /* *EyeSizeRadius_Return=-99.9; */
```

```
*RadiusMaxWind_Return=MISSFN1;
*EyeSizeRadius_Return=MISSFN1;
/* iterate five times */
for(Iteration=0;Iteration<5;Iteration++) {
 XInc=CenterPointXDir;
 while(IRDataGrid_Global->temp[CenterPointYDir][XInc]>
   CriticalTemperatureDegK) {
  XInc=XInc-1;
  if(XInc==XDirMinimum) {
   /* Eyewall not found */
   return -1;                  /* EXIT */
  }
 }
 XDirIterationStoredMinimum=XInc;
 XInc=CenterPointXDir;
 while(IRDataGrid_Global->temp[CenterPointYDir][XInc]>
   CriticalTemperatureDegK) {
  XInc=XInc+1;
  if(XInc==XDirMaximum) {
   /* Eyewall not found */
   return -1;                  /* EXIT */
  }
 }
 XDirIterationStoredMaximum=XInc;
 YInc=CenterPointYDir;
 while(IRDataGrid_Global->temp[YInc][CenterPointXDir]>
   CriticalTemperatureDegK) {
  YInc=YInc-1;
  if(YInc==YDirMinimum) {
   /* Eyewall not found */
   return -1;                  /* EXIT */
  }
 }
 YDirIterationStoredMinimum=YInc;
 YInc=CenterPointYDir;
 while(IRDataGrid_Global->temp[YInc][CenterPointXDir]>
   CriticalTemperatureDegK) {
  YInc=YInc+1;
  if(YInc==YDirMaximum) {
   /* Eyewall not found */
   return -1;                  /* EXIT */
  }
 }
 YDirIterationStoredMaximum=YInc;
 CenterPointXDir=(int)((((float)(XDirIterationStoredMinimum+
            XDirIterationStoredMaximum))/2.0));
 CenterPointYDir=(int)((((float)(YDirIterationStoredMinimum+
            YDirIterationStoredMaximum))/2.0));
}
CenterPointLatitude=
     IRDataGrid_Global->lat[CenterPointYDir][CenterPointXDir];
CenterPointLongitude=
     IRDataGrid_Global->lon[CenterPointYDir][CenterPointXDir];
LatitudeValue=
     IRDataGrid_Global->lat[CenterPointYDir][XDirIterationStoredMinimum];
LongitudeValue=
     IRDataGrid_Global->lon[CenterPointYDir][XDirIterationStoredMinimum];
aodt_distance(LatitudeValue,LongitudeValue,
          CenterPointLatitude,CenterPointLongitude,1,
          &DistanceValueX1,&AngleReturnValue);
LatitudeValue=
     IRDataGrid_Global->lat[CenterPointYDir][XDirIterationStoredMaximum];
LongitudeValue=
     IRDataGrid_Global->lon[CenterPointYDir][XDirIterationStoredMaximum];
aodt_distance(LatitudeValue,LongitudeValue,
          CenterPointLatitude,CenterPointLongitude,1,
          &DistanceValueX2,&AngleReturnValue);
LatitudeValue=
     IRDataGrid_Global->lat[YDirIterationStoredMinimum][CenterPointXDir];
LongitudeValue=
     IRDataGrid_Global->lon[YDirIterationStoredMinimum][CenterPointXDir];
aodt_distance(LatitudeValue,LongitudeValue,
          CenterPointLatitude,CenterPointLongitude,1,
          &DistanceValueY1,&AngleReturnValue);
LatitudeValue=
     IRDataGrid_Global->lat[YDirIterationStoredMaximum][CenterPointXDir];
LongitudeValue=
     IRDataGrid_Global->lon[YDirIterationStoredMaximum][CenterPointXDir];
aodt_distance(LatitudeValue,LongitudeValue,
          CenterPointLatitude,CenterPointLongitude,1,
          &DistanceValueY2,&AngleReturnValue);
AveragedDistance=(DistanceValueX1+DistanceValueX2+
            DistanceValueY1+DistanceValueY2)/4.0;
if(AveragedDistance>0.0) {
  *RadiusMaxWind_Return=2.8068+(0.8361*AveragedDistance);
  *EyeSizeRadius_Return=AveragedDistance;
}
 return Ret_ID;
}
```

8.4 Forecast Interpolation routine

```
{
int XInc;                        /* X direction increment */
int YInc;                        /* Y direction increment */
int StoreValue;                  /* storage value */
float LocalArray1[5];            /* local analysis array */
float LocalArray2[5];            /* local analysis array */
float TimeDifferenceValue1;         /* time difference between positions */
float TimeDifferenceValue2;         /* time difference between positions */
float TimeValue1;                /* time value of first point */
float TimeValue2;                /* time value of second point */
float PositionValue;             /* position value */
float DenominatorValue;            /* denominator value */
float PositionError_Local;         /* local position error value */
float PositionValue_Local;         /* local position value */

StoreValue=1;
TimeDifferenceValue1=A_ABS(CurrentTime-TimeArrayInput[0]);
for(XInc=1;XInc<=NumberOfPointsInput;XInc++) {
  TimeDifferenceValue2=A_ABS(CurrentTime-TimeArrayInput[XInc-1]);
  if(TimeDifferenceValue2<TimeDifferenceValue1) {
    StoreValue=XInc;
    TimeDifferenceValue1=TimeDifferenceValue2;
  }
  LocalArray1[XInc-1]=PositionArrayInput[XInc-1];
  LocalArray2[XInc-1]=PositionArrayInput[XInc-1];
}
PositionValue_Local=PositionArrayInput[StoreValue-1];
StoreValue--;
for(YInc=1;YInc<=NumberOfPointsInput-1;YInc++) {
  for(XInc=1;XInc<=NumberOfPointsInput-YInc;XInc++) {
    TimeValue1=TimeArrayInput[XInc-1]-CurrentTime;
    TimeValue2=TimeArrayInput[(XInc-1)+YInc]-CurrentTime;
    PositionValue=LocalArray1[XInc]-LocalArray2[XInc-1];
    DenominatorValue=TimeValue1-TimeValue2;
    if(DenominatorValue==0.0) {
      /* interpolation error - denominator = 0 */
      return -1;                        /* EXIT */
    }
    DenominatorValue=PositionValue/DenominatorValue;
    LocalArray1[XInc-1]=TimeValue1*DenominatorValue;
    LocalArray2[XInc-1]=TimeValue2*DenominatorValue;
  }
```

```
      if((2*StoreValue)<(NumberOfPointsInput-YInc)) {
        PositionError_Local=LocalArray1[StoreValue];
      }
      else {
        PositionError_Local=LocalArray2[StoreValue-1];
        StoreValue--;
      }
      PositionValue_Local=PositionValue_Local+PositionError_Local;
    }
    *InterpolatedPositionError_Return=PositionError_Local;
    *InterpolatedPosition_Return=PositionValue_Local;
     return 0;
   }
```

8.5 Rectilinear Remapping discussion

A high-level discussion of the rectilinear remapping description is provided in Section 3.4.2.1. The process of filling the destination array is done through spline boxes, ElementSplineInput by LineSplineInput in size. The Line[UL,UR,LL,LR]Value and Element[UL,UR,LL,LR]Value arrays contain the line and element locations of the source pixels to put at the corners of the spline boxes. The boxes are square in the destination array; however, that box reprojected onto the source data array will be an arbitrary polygon.

The following is edited from the original code to highlight the important parts:
Initially a check is done on the source, interpolation or temporary buffers. If any of those are null, the function sends a return code of -1, which is designated as an array buffer error.

```
  if((SourceCharBuffer==NULL)||(InterpCharBuffer==NULL)||
      (TempCharBuffer==NULL)) {
      Ret_ID=-1;
    }
```

If the source, interpolation or temporary buffers are not null, the function continues with the filling of the destination array.

The Spline pixel value is initialized to the product of the spline function line and element value. The next step is to fill the temporary buffer (TempCharBuffer) with a 0 value. After this is performed each 1-d index of the source array is initially filled with the temp field at a given line and element, from the IRDataGridGlobal pointer, as shown below. The index of the Source array is determined by the line and element, as shown below.

```
  for(YInc= 0;YInc<tiff_vars.in_lines;YInc++) {
      for(XInc=0;XInc<tiff_vars.in_elems;XInc++) {
        ArrayIndexValue=(YInc*tiff_vars.in_elems)+XInc;
        SourceArray[ArrayIndexValue]=IRDataGrid_Global->temp[YInc][XInc];
      }
    }
```

Recall that tiff_vars.in_lines is the Maximum Latitude variable minus the Minimum Latitude variable, all divided by the latitude increment ((MaxLat – MinLat) / LatInc). Similarly tiff_vars.out_elems is the maximum longitude variable minus the minimum longitude variable, divided by the longitude increment, with according checks on if the dateline is crossed. These variables are determined in the aodt_determinedest function, previously described.

After the source array is initialized, the Line and Element coordinates for the previous 1D element are initialized in the following loop.

```
      for(YInc=1;YInc<remap_vars.nce+1;YInc++) {
        for(XInc=1;XInc<remap_vars.ncl+1;XInc++) {
```

Where remap_vars.nce and remap_vars.ncl are initialized in the aodt_int function.

For each line and element pair, the index in the array (ArrayPointValue) is set as the value of IND at (YInc,XInc)

If the value of the line coordinates to be transformed (LineCoordsArrayInput) at index (ArrayPointValue – 1) is equal to the missing value sentinel MISSFN, then the Line coordinate, element coordinate and temporary character arrays at index (ArrayPointValue – 1) must be determined. This is done as follows:

- Initialize Line sum (LineSum), Element sum (ElementSum) and pointer counter (PointerCounter) to 0
- For increment value (IncVal) 0 to 7
    - YInc1=YInc + LineIncArray1[IncVal]
        - LineIncArray1 = {-2,-2, 0, 2, 2, 2, 0,-2}
    - YInc2=YInc + LineIncArray2[IncVal]
        - LineIncArray2 = {-1,-1, 0, 1, 1, 1, 0,-1}
    - Bounds check Yinc1 and Yinc2
        - If Yinc1 < 1, YInc2 <1, Yinc1 > remap_vars.ncl or Yinc2 > remap_vars.ncl, skip to next increment
    - Calculate element increment in similar manner
        - XInc1=XInc+ElementIncArray1[IncVal]
            - ElementIncArray1 = { 0, 2, 2, 2, 0,-2,-2,-2}
        - XInc2=XInc+ElementIncArray2[IncVal]
            - ElementIncArray2 = { 0, 1, 1, 1, 0,-1,-1,-1}
    - Bounds check element increment

        - If Xinc1 < 1, XInc2 <1, Xinc1 > remap_vars.nce or Xinc2 > remap_vars.nce, skip to next increment
    - Set ValueIND1 to IND(YInc1, XInc1)
    - Set ValueIND2 to IND(YInc2, XInc2)
    - If either, the input line coordinate array (LineCoordsArrayInput) at index (ValueIND1 -1) or at index (ValueIND2 -1) is equal to MISSFN1, then skip to next increment.
    - If the temporary buffer at index (ValueIND1 -1) or at index (ValueIND2 - 1) is **not** equal to 0 then skip to next increment
    - Otherwise
        - add 1 to PointCounter
        - LineSum=LineSum + ((float)(2.0*LineCoordsArrayInput[ValueIND2-1]))- LineCoordsArrayInput[ValueIND1-1]
        - ElementSum=ElementSum + ((float)2.0*ElementCoordsArrayInput[ValueIND2-1])- ElementCoordsArrayInput[ValueIND1-1]
- If PointCounter is greater than 0, then we set the Line and element coordinate array as well as the temporary character buffer at index (ArrayPointValue-1) as follows
    - LineCoordsArrayInput[ArrayPointValue-1]=LineSum / PointCounter
    - ElementCoordsArrayInput[ArrayPointValue-1]=ElementSum / PointCounter
    - TempCharBuffer[ArrayPointValue-1] = 1

It is assumed that if LineCoordsArrayInput(ArrayPointValue – 1) /= missing that the values of the Line coordinate, element coordinate and temporary character arrays at index (ArrayPointValue – 1) have been determined

Once the values of the Line coordinate, element coordinate and temporary character arrays at index (ArrayPointValue – 1) have been determined, the actual rectilinear correction is performed. First BlockCounter is initialized to 0. The outer loop increments IncVal in the following manner.

```
for(IncVal=1;IncVal<tiff_vars.out_lines+1;IncVal=IncVal+remap_vars.dlb) {
```

For each increment through the 1D array, the accumulated lines per block for each increment (AccumulatedLines) is defined as the product of BlockCounter (i.e. the number of blocks processed) * remap_vars.dlb.

The pointer which holds the first corner (FirstCornerPointer) is set equal to the product of BlockCounter * remap_vars.ncl *remap_vars.dlb, all divided by the spline function line value (LineSplineInput). This result is then also put temporarily in a holding variable (FirstCornerPointerHold).

The last corner of the destination block (LastCornerPointer) is set equal to (BlockCounter + 1) * remap_vars.ncl *remap_vars.dlb, all divided by the spline function line value (LineSplineInput), then subtracted by 1. The final result of LastCornerPointer is set to the minimum of result of the previous sentence or the total number of points – remap_vars.ncl, using the A_MIN function (LastCornerPointer=A_MIN(LastCornerPointer, NumberOfCornersInput-remap_vars.ncl

Now that the first and last corner are initialized, the function now loops through the entire source block and then loops through each spline, moving any data as necessary.

The first thing that is done in the spline loop is to set the offset in the destination array for the beginning of the spline boxes, OffsetValue0. For the first row of boxes, this will be zero. For the 2nd row of boxes, it will be set to the LineSplineInput*ElementSplineInput. So, if the spline is 10 and there are 100 elements in the line, then OffsetValue0 = 1000 for the 2nd row of boxes.

OffsetValue1 is the offset into row. It is zero at the beginning of the line. It is espline when working on the 2nd box in a line.

```
OffsetValue0=(SourceBlockValue/remap_vars.nspl)*
          LineSplineInput*tiff_vars.out_elems;
OffsetValue1=(SourceBlockValue%remap_vars.nspl)*
          ElementSplineInput;
OffsetValue2=OffsetValue1+OffsetValue0+1;
CornerPointer=FirstCornerPointer+SplineInc-1;
```

After these values are initialized, the 4 corners in line spaced are determined and checked to see if they are out of bounds. The ordering is as follows

1,1(UL)	**1,2 (UR)**
2,1(LL)	**2,2 (LR)**

The x direction value (ValX) is set to remap_vars.ncl and the upper left, upper right and lower left corners are determined as follows

```
LineULValue=LineCoordsArrayInput[CornerPointer];
LineURValue=LineCoordsArrayInput[CornerPointer+1];
LineLLValue=LineCoordsArrayInput[CornerPointer+ValX];
```

After they have been determined, the lower right line value is determined. This is done by looking to see if the pointer to the corner (CornerPointer) plus the x direction value plus one is greater than the number of points to be transformed, then the line lower right corner (LineLRValue) is set to LineCoordsArrayInput[CornerPointer+1+ValX]. Otherwise, LineLRValue=LineCoordsArrayInput[CornerPointer+ValX].

After the four corners are obtained, then minimum line value (LineMinimum) is obtained through a series of checks of each corner

```
LineMinimum=A_MIN(LineULValue,LineURValue);
LineMinimum=A_MIN(LineMinimum,LineLLValue);
LineMinimum=A_MIN(LineMinimum,LineLRValue);
```

If a limb exists within the spline box, (i.e LineMinimum == (float)MISSFN1), then the rest of the process is skipped by going to LABEL30, shown below.

If there is no limb present in the spline box, the minmum Y direction value (LineMinimumFinal) is set to LineMinimum + 0.5. If this value is greater than the maximum source array line (MaximumSourceLine), the rest of the process is skipped by going to LABEL30, shown below.

If LineMinimumFinal is less than or equal to MaximumSourceLine, then the maximum line value (LineMaximum) is determined

```
LineMaximum=A_MAX(LineULValue,LineURValue);
LineMaximum=A_MAX(LineMaximum,LineLLValue);
LineMaximum=A_MAX(LineMaximum,LineLRValue);
```

The maximum Y direction value (LineMaximumFinal) is the sum of LineMaximum and 0.5. If this value is less than the source array increment (SourceBlockInc), then the rest of the process is skipped by going to LABEL30. Otherwise, a check of the elements of each of the 4 corners is performed in a similar manner to the line of each

```
    /* Get 4 corners in elem space & check for out of bound */

ValX=remap_vars.ncl;
ElementULValue=ElementCoordsArrayInput[CornerPointer];
ElementURValue=ElementCoordsArrayInput[CornerPointer+1];
ElementLLValue=ElementCoordsArrayInput[CornerPointer+ValX];
if((CornerPointer+1+ValX)<NumberOfCornersInput) {
  ElementLRValue=ElementCoordsArrayInput[CornerPointer+1+ValX];
}
else {
  ElementLRValue=ElementCoordsArrayInput[CornerPointer+ValX];
}
Determining ElementMaximumFinal
ElementMaximum=A_MAX(ElementULValue,ElementURValue);
ElementMaximum=A_MAX(ElementMaximum,ElementLLValue);
ElementMaximum=A_MAX(ElementMaximum,ElementLRValue);
```

```
ElementMaximumFinal=(int)(ElementMaximum+(float)0.5);
if(ElementMaximumFinal<1) {
  goto LABEL30;
}
Determining ElementMinimumFinal
ElementMinimum=A_MIN(ElementULValue,ElementURValue);
ElementMinimum=A_MIN(ElementMinimum,ElementLLValue);
ElementMinimum=A_MIN(ElementMinimum,ElementLRValue);
ElementMinimumFinal=(int)(ElementMinimum+(float)0.5);

if(ElementMinimumFinal>tiff_vars.in_elems) {
  goto LABEL30;
}
```

Initially, the array edge flag (EdgeFixID) is set to 0. If the final element falls off the edge of the image (ElementMaximumFinal > tiff_vars_in_elems AND ElementMinimumFinal is less than 1), the pixel is pitched by skipping to LABEL30

LineULValue, LineURValue, LineLLValue, LineLRValue contain the line numbers (likewise, ElementULValue, etc. for elements) for the pixels to retrieve from the source polygon.

After that is done, the determination (i.e. question 2) of if the pixel is on the edge of an array (EdgeFixID) is performed. This is done by checking to see if the right and left edges are continuous.

```
/* Fix if left & right edge should be continuous */
Criteria for making EdgeFixID = 1
if((ElementMaximumFinal-ElementMinimumFinal)>
  (int)(.75*tiff_vars.in_elems)) {
  if(ElementULValue<(tiff_vars.in_elems/2)) {
    ElementULValue=ElementULValue+tiff_vars.in_elems;
  }
  if(ElementURValue<(tiff_vars.in_elems/2)) {
    ElementURValue=ElementURValue+tiff_vars.in_elems;
  }
  if(ElementLLValue<(tiff_vars.in_elems/2)) {
    ElementLLValue=ElementLLValue+tiff_vars.in_elems;
  }
  if(ElementLRValue<(tiff_vars.in_elems/2)) {
    ElementLRValue=ElementLRValue+tiff_vars.in_elems;
  }

  EdgeFixID=1;
}
```

The dimension of the destination spline box to fill is size LineSplineInput by ElementSplineInput.
The pixels within the source polygon are referenced in terms of the starting line,elem and the delta line,elem per pixel.

This is done so as we traverse the spline box, pixel by pixel, these deltas are added in the source polygon. LineValueA is the fractional change in source line number from 1,1 to 1,2; LineValueB is the fractional change in source line number from 1,1 to 2,1; etc.

The calculation of the fractional change is in source line/element is shown below:

```
LineValueA=(LineURValue-LineULValue)/ElementSplineInput;
LineValueB=(LineLLValue-LineULValue)/LineSplineInput;
LineValueC=(LineLRValue+LineULValue-LineURValue-LineLLValue)/
        SplinePixelValue;
ElementValueA=(ElementURValue-ElementULValue)/
          ElementSplineInput;
ElementValueB=(ElementLLValue-ElementULValue)/
          LineSplineInput;
ElementValueC=(ElementLRValue+ElementULValue-
          ElementLLValue-ElementURValue)/
          SplinePixelValue;
```

LineValueBCounter is the ZB source line number term; ElementValueBCounter is the ZB source element number term.

At the beginning of a spline box, set to first source line, LineValueBCounter, and first source element, ElementValueBCounter as follows:

```
ElementMiscValue=0;
LineValueBCounter=LineULValue+(float)0.5;
LineValueCCounter=(float)0.0;
ElementValueBCounter=ElementULValue+(float)0.5;
ElementValueCCounter=(float)0.0;
```

If one of the following conditions is met:

- SplineInc==remap_vars.nspl
- (ElementMinimumFinal<1 or ElementMaximumFinal>tiff_vars.in_elems)
- (LineMinimumFinal<SourceBlockInc) or (LineMaximumFinal>MaximumSourceLine)
- (FirstCornerPointer+(2*remap_vars.ncl)-1) greater than LastCornerPointer)

One takes processing path 1. If none of the conditions are met, then one takes processing path 2

Processing Path 1

One goes through each line

```
                for(YInc=1;YInc<LineSplineInput+1;YInc++) {
```

Initializing LineValueACounter, the delta line from both the ZA term and the ZC term, ElementValueACounter, the delta elem from both the ZA term and the ZC term, LineValueACounter0, the accumulated value of LineValueACounter going across a line, and the ElementValueACounter0, the accumulated value of ElementValueACounter going across a line, as follows.

```
                  LineValueACounter=LineValueCCounter+LineValueA;
                  ElementValueACounter=ElementValueCCounter+ElementValueA;
                  LineValueACounter0=(float)0.0;
                  ElementValueACounter0=(float)0.0;
```

Within each line of the destination spline box, we then go through element-by-element

```
                        for(XInc=1;XInc<ElementSplineInput+1;XInc++) {
```

And then determine the source line (SourceLine) by adding the ZA, ZB, ZC terms together [each is already multiplied by the appropriate delta], (LineValueBCounter + LineValueACounter0)

If the SourceLine is less than the source block increment (SourceBlockInc) or is greater than the maximum source line (MaximumSource Line), then the function skips to LABEL38, shown below.

If neither bounds check for the source line is triggered, then the source element (SourceElement) is determined by adding (ElementValueBCounter + ElementValueACounter0). One then checks the bounds on SourceElement. If SourceElement is less than 1 or if (SourceElement > tiff_vars.in_elems and EdgeFixID == 0) then skip to LABLE38.

If SourceElement is within bounds, then the function determines the source 1D pixel offset. Since the arrays are small, assume that sloc=1. Then the equation becomes the typical way to determine the 1D index in a 2D array:
index=(line – 1) * number_elems + element

```
                          SourceOffsetValue=((SourceLine-SourceBlockInc)*
                                tiff_vars.in_elems)+SourceElement;
```

The 1D offsets are determined and checked through the following manner

```
OffsetValueE=OffsetValue1+XInc;
Bounds checking for OffsetValueE
if(OffsetValueE>tiff_vars.out_elems) {
      goto LABEL38;
}
OffsetValueL=OffsetValue0+ElementMiscValue;

Bounds checking for OffsetValueL
if((OffsetValueL/
     (tiff_vars.out_elems+AccumulatedLines-1))>
     tiff_vars.out_lines) {
     goto LABEL38;
}

OffsetValue=OffsetValueL+OffsetValueE;
```

Fill the destination array with appropriate pixel from source array

```
     InterpArray[OffsetValue-1]=
     SourceArray[SourceOffsetValue-1];
```

Below the location where InterpArray[OffsetValue -1] is where LABEL38 is located. This is where all the goto statements go to.

At this point, we need to increment the zlac0 term only by the ZC term, since the delta line does not change

```
 LineValueACounter0=LineValueACounter0+
      LineValueACounter;
 ElementValueACounter0=ElementValueACounter0+
      ElementValueACounter;
     }
```

To advance to next line in source polygon, add the delta line LineValueB, delta element ElementValueB, and delta bilinear terms LineValueC and ElementValueC

```
LineValueBCounter=LineValueBCounter+LineValueB;
LineValueCCounter=LineValueCCounter+LineValueC;
ElementValueBCounter=ElementValueBCounter+
               ElementValueB;
ElementValueCCounter=ElementValueCCounter+
               ElementValueC;
ElementMiscValue=ElementMiscValue+tiff_vars.out_elems;               }

}
```

```
            }
```

Processing Path 2

Different processing paths to obtain LineValueBCounter, LineValueCCounter, ElementValueBCounter, and ElementValueCCounter depending on any fixes are made at the edges

```
        else {
            if(EdgeFixID==0) {
        “”’’
        }
            }
            else if(EdgeFixID==1) {
        “”’’
        }
        LABEL30:
       }
     }
   }
 }
```

The processing path if EdgeFixID is 1 is nearly identical to the EdgeFix ==0 case. The only execption is that in the EdgeFixID==1 case, there is a check on the source array element value(SourceElement). If SourceElement > tiff_vars.in_elems, then tiff_vars.in_elems is subtracted from SourceElement.

After either processing path 1 or 2 are taken, the final results are placed into the appropriate locations of the IRDataGridRemap structure

```
 for(YInc=0;YInc<tiff_vars.out_lines;YInc++) {
      for(XInc=0;XInc<tiff_vars.out_elems;XInc++) {
        ArrayIndexValue=(YInc*tiff_vars.out_elems)+XInc;
        IRDataGridRemap->temp[YInc][XInc]=InterpArray[ArrayIndexValue];
      }
    }

  }
}
```

If at the end, a return code of 0 is returned to the function.

8.6 Fast Fourier Transform (FFT) algorithm

```
int aodt_dfft( double RealArray_Input[2],
            double CmplxArray_Input[2],
            int NumBins )
```

```
/*
** A Duhamel-Hollman split-radix dif fft
** Ref: Electronics Letters, Jan. 5, 1984
** Complex input and output data in arrays x and y
** Length is n.
** Inputs  : RealArray_Input  - Input data array to perform FFT analysis
**          CmplxArray_Input - Empty on input
**          NumBins          - Number of histogram bins in input array
** Outputs : RealArray_Input  - Real part of FFT Analysis
**          CmplxArray_Input - Complex part of FFT Analysis
** Return  : <=0- error; >0 - o.k.
*/
{
 double *RealArr;                    /* real values array */
 double *CmplxArr;                   /* complex values array */
 int LocalA;                         /* local variable */
 int LocalB;                         /* local variable */
 int LocalC;                         /* local variable */
 int LocalD;                         /* local variable */
 int LocalE;                         /* local variable */
 int LocalX0;                        /* local variable */
 int LocalX1;                        /* local variable */
 int LocalX2;                        /* local variable */
 int LocalX3;                        /* local variable */
 int LocalY;                         /* local variable */
 int LocalZ;                         /* local variable */
 int LocalE1;                        /* local variable */
 int LocalE2;                        /* local variable */
 int LocalE4;                        /* local variable */
 double LocalDblA;                   /* local variable */
 double LocalDblB;                   /* local variable */
 double LocalDblA3;                  /* local variable */
 double LocalDblX1;                  /* local variable */
 double LocalDblX3;                  /* local variable */
 double LocalDblY1;                  /* local variable */
 double LocalDblY3;                  /* local variable */
 double LocalDblW1;                  /* local variable */
 double LocalDblW2;                  /* local variable */
 double LocalDblZ1;                  /* local variable */
 double LocalDblZ2;                  /* local variable */
 double LocalDblZ3;                  /* local variable */
 double LocalDblXt;                  /* local variable */

 RealArr = RealArray_Input - 1;
 CmplxArr = CmplxArray_Input - 1;
 LocalA = 2;
```

```
LocalD = 1;
while (LocalA < NumBins) {
  LocalA = LocalA+LocalA;
  LocalD = LocalD+1;
}
LocalE = LocalA;
if (LocalE != NumBins) {
  for (LocalA = NumBins+1; LocalA <= LocalE; LocalA++)  {
    *(RealArr + LocalA) = 0.0;
    *(CmplxArr + LocalA) = 0.0;
  }
}

LocalE2 = LocalE+LocalE;
for (LocalC = 1;  LocalC <= LocalD-1; LocalC++ ) {
  LocalE2 = LocalE2 / 2;
  LocalE4 = LocalE2 / 4;
  LocalDblB = 2.0 * PI / LocalE2;
  LocalDblA = 0.0;
  for (LocalB = 1; LocalB<= LocalE4 ; LocalB++) {
    LocalDblA3 = 3.0*LocalDblA;
    LocalDblX1 = cos(LocalDblA);
    LocalDblY1 = sin(LocalDblA);
    LocalDblX3 = cos(LocalDblA3);
    LocalDblY3 = sin(LocalDblA3);
    LocalDblA = ((double)LocalB)*LocalDblB;
    LocalY = LocalB;
    LocalZ = 2*LocalE2;
    while ( LocalY < LocalE ) {
      for (LocalX0 = LocalY; LocalX0 <= LocalE-1;
        LocalX0 = LocalX0 + LocalZ) {
        LocalX1 = LocalX0 + LocalE4;
        LocalX2 = LocalX1 + LocalE4;
        LocalX3 = LocalX2 + LocalE4;
        LocalDblW1 = *(RealArr+LocalX0) - *(RealArr+LocalX2);
        *(RealArr+LocalX0) = *(RealArr+LocalX0) + *(RealArr+LocalX2);
        LocalDblW2 = *(RealArr+LocalX1) - *(RealArr+LocalX3);
        *(RealArr+LocalX1) = *(RealArr+LocalX1) + *(RealArr+LocalX3);
        LocalDblZ1 = *(CmplxArr+LocalX0) - *(CmplxArr+LocalX2);
        *(CmplxArr+LocalX0) = *(CmplxArr+LocalX0) + *(CmplxArr+LocalX2);
        LocalDblZ2 = *(CmplxArr+LocalX1) - *(CmplxArr+LocalX3);
        *(CmplxArr+LocalX1) = *(CmplxArr+LocalX1) + *(CmplxArr+LocalX3);
        LocalDblZ3 = LocalDblW1 - LocalDblZ2;
        LocalDblW1 = LocalDblW1 + LocalDblZ2;
        LocalDblZ2 = LocalDblW2 - LocalDblZ1;
        LocalDblW2 = LocalDblW2 + LocalDblZ1;
```

```
          *(RealArr+LocalX2) = LocalDblW1*LocalDblX1 -
                        LocalDblZ2*LocalDblY1;
          *(CmplxArr+LocalX2) = -LocalDblZ2*LocalDblX1 -
                         LocalDblW1*LocalDblY1;
          *(RealArr+LocalX3) = LocalDblZ3*LocalDblX3 +
                        LocalDblW2*LocalDblY3;
          *(CmplxArr+LocalX3) = LocalDblW2*LocalDblX3 -
                         LocalDblZ3*LocalDblY3;
        }
        LocalY = 2*LocalZ - LocalE2 + LocalB;
        LocalZ = 4*LocalZ;
      }
    }
  }

/*
---------------------Last stage, length=2 butterfly---------------------
*/
  LocalY = 1;
  LocalZ = 4;
  while ( LocalY < LocalE) {
    for (LocalX0 = LocalY; LocalX0 <= LocalE; LocalX0 = LocalX0 + LocalZ) {
      LocalX1 = LocalX0 + 1; LocalDblW1 = *(RealArr+LocalX0);
      *(RealArr+LocalX0) = LocalDblW1 + *(RealArr+LocalX1);
      *(RealArr+LocalX1) = LocalDblW1 - *(RealArr+LocalX1);
      LocalDblW1 = *(CmplxArr+LocalX0);
      *(CmplxArr+LocalX0) = LocalDblW1 + *(CmplxArr+LocalX1);
      *(CmplxArr+LocalX1) = LocalDblW1 - *(CmplxArr+LocalX1);
    }
    LocalY = 2*LocalZ - 1;
    LocalZ = 4 * LocalZ;
  }

/*
c--------------------------Bit reverse counter
*/
  LocalB = 1;
  LocalE1 = LocalE - 1;
  for (LocalA = 1; LocalA <= LocalE1; LocalA++) {
    if (LocalA < LocalB) {
      LocalDblXt = *(RealArr+LocalB);
      *(RealArr+LocalB) = *(RealArr+LocalA);
      *(RealArr+LocalA) = LocalDblXt;
      LocalDblXt = *(CmplxArr+LocalB);
      *(CmplxArr+LocalB) = *(CmplxArr+LocalA);
      *(CmplxArr+LocalA) = LocalDblXt;
    }
    LocalC = LocalE / 2;
    while (LocalC < LocalB) {
      LocalB = LocalB - LocalC;
      LocalC = LocalC / 2;
    }
    LocalB = LocalB + LocalC;
  }

  return LocalE;

}
```